3-14-2008

Modeling GPS Satellite Orbits Using KAM Tori

Rachel M. Derbis

Modeling GPS Satellite Orbits
Using KAM Tori

THESIS

Rachel M. Derbis, Captain, USAF

AFIT/GA/ENY/08-M09

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GA/ENY/08-M09

Modeling GPS Satellite Orbits Using KAM Tori

THESIS

Presented to the Faculty

Department of Aeronautics and Astronautics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Astronautical Engineering

Rachel M. Derbis, BS

Captain, USAF

March 2008

AFIT/GA/ENY/08-M09

Modeling GPS Satellite Orbits Using KAM Tori

Rachel M. Derbis, BS

Captain, USAF

Approved:

William E. Wiesel	14 Mar 08
William E. Wiesel, PhD (Chairman)	Date
	14 Mar 08
LtCol Kerry D. Hicks, PhD (Member)	Date
	14 Mar 08
LtCol Nathan A. Titus, PhD (Member)	Date

Abstract

Global Positioning System (GPS) satellite orbits are modeled using Kolmogorov, Arnold, Moser (KAM) tori. Precise Global Positioning System satellite locations are analyzed using Fourier transforms to identify the three basis frequencies in an Earth Centered, Earth Fixed (ECEF) rotating reference frame. The three fundamental frequencies are 1) the anomalistic frequency, 2) a combination of earth's rotational frequency and the nodal regression rate, and 3) the apsidial regression rate. A KAM tori model fit to the satellite data could be used to predict future satellite locations. This model would allow rapid determination with fewer computational requirements than the typical method of integrating through an orbit.

Acknowledgements

I'd like to thank my husband, who has been extremely supportive throughout my graduate studies. A few individuals have helped me with different sections of code. I am grateful to Rich, Olek, and Matt for their assistance in this area. I'd also like to thank my mother for all her help proof reading my thesis. Last, but certainly not least, I would like to thank my Committee Chairman, Dr William Wiesel, who acted as a research advisor and the committee members who instructed me in the many Astrodynamics classes I have taken during my graduate studies at the Air Force Institute of Technology.

Rachel M. Derbis

Table of Contents

Page

List of Figures

Page

List of Tables

List of Symbols

Symbol		Page
$R_\oplus$	mean radius of the Earth	14
μ	Earth gravitational parameter	14
J_2	J_2 term of the geopotential	14
$\omega_\oplus$	Earth rotational frequency	14
e	orbit eccentricity	14
a	orbit semi-major axis	14
i	orbit inclination	14
$\mathbf{X}$	state matrix of satellite	14

List of Abbreviations

Modeling GPS Satellite Orbits Using KAM Tori

I. Introduction

Since the launch of Sputnik on October 4, 1957, the number of objects orbiting the earth has increased. These objects include commercial and military satellites, spacecraft, and debris. These objects must be tracked to reduce the risk of hypervelocity impacts. Currently, the US Space Surveillance Network is tracking over 12,000 objects in Earth orbit.

1.1 Motivation

The current methods for predicting the location of tracked satellites are based on integrating Kepler's equations and assuming small perturbations. These methods are time and computer intensive. Another method used over short periods of time is to estimate a satellite's future trajectory by projecting the average of the recent trajectory forward. The Global Positioning System (GPS) uses this method to provide more accurate solutions for satellite location, although the predictions are valid for only a few hours. Radar and telescopes are used to determine the precise position and to confirm the predicted location of an object orbiting earth. If an object's location changes due to space weather effects or a maneuver, it must be reacquired by tracking systems and a new orbit must be calculated for the new trajectory of the object. After a major event such as the geomagnetic storm of March 1989, thousands of earth orbiting satellites can be "lost". It can take days to begin tracking all of the objects again.

A new method allowing direct prediction of a satellite's location at any point in time would allow tracking of additional objects without requiring additional resources. It would also make a file with satellite location calculation parameters valid for a longer period of time.

1.2 Background

Current methods for predicting satellite positions are computationally intensive and take significant amounts of time. A method for directly calculating a satellite location at any point in time would be beneficial.

1.3 Approach

The Kolmogorov, Arnold, Moser (KAM) theory states that if a trajectory only has small perturbations to the Hamiltonian, then it will lie on a torus. This torus is represented by a Fourier series with the same number of frequencies as the coordinates of the system. A Fast Fourier Transform (FFT) is completed on orbit data to determine if it has discrete frequencies, and if so, what those frequencies are.

1.4 Problem Statement

This thesis applies the KAM theorem to precise satellite data from the GPS satellites. Showing that the orbits have a distinct set of frequencies illustrates that the orbits lie on tori.

1.5 Results

Analysis shows that GPS satellites follow the KAM theory, having three distinct frequencies. Some of the older satellites whose orbits have started to decay show semi-stable frequency mappings. Further analysis is required to fit the coefficients to an orbital model. These results could then be verified by calculating positions and comparing the calculated positions with actual data.

II. Background

This chapter begins with a discussion of the technical characteristics of GPS relative to the analysis that follows. Current orbit modeling capabilities are provided as background to understand how changing the modeling method can increase the overall capability to establish satellite locations. The KAM theorem, the basis for this thesis, is discussed to understand the methods used for the analysis in Chapters III and IV. Finally, this chapter provides a review of literature relative to space object applications of the KAM theorem done by other researchers.

2.1 Global Positioning System

Initial operational capability for the GPS was obtained in December 1993. The satellites are in semi-synchronous near circular orbits with a period of 11 hours and 58 minutes per orbit. The semi-major axis for each orbit is 26,560 km. The nominal constellation configuration consists of at least 24 satellites with four satellites arranged in each of six orbital planes. In 2007, 31 satellites were in operation for some part of the year [*Milcom Monitoring Post*, 2007] [NGIA, 2008] . Each orbital plane has an inclination of 55°. The Right Ascension of the Ascending Node (RAAN) for the orbital planes are as follows A) 272.85° B) 332.85° C) 32.85° D) 92.85° E) 152.85° F) 212.85° [*Misra and Enge*, 2001]. Figure 2.1 below was generated using Satellite Tool Kit (STK) version 8.1. It shows the GPS satellites that were in orbit on 1 January 2007. This figure illustrates the six orbital planes and the satellites spaced out within each of the planes.

The international standard time is coordinated universal time (UTC). Universal time has days equal to the mean solar day and includes the irregularities in the Earth's rotation. UTC is maintained to within 0.9 seconds of universal time through the use of leap seconds. GPS time was set to match UTC on 6 January 1980. GPS time does not include leap seconds and therefore UTC is currently 14 seconds faster than GPS time. Receivers must take this difference into account when they calculate their time in UTC.

All GPS satellites publish an almanac which provides the approximate ephemeris data with orbital elements for all of the satellites. Receivers use this almanac data to acquire satellites. Each individual satellite transmits it's broadcast ephemeris data and

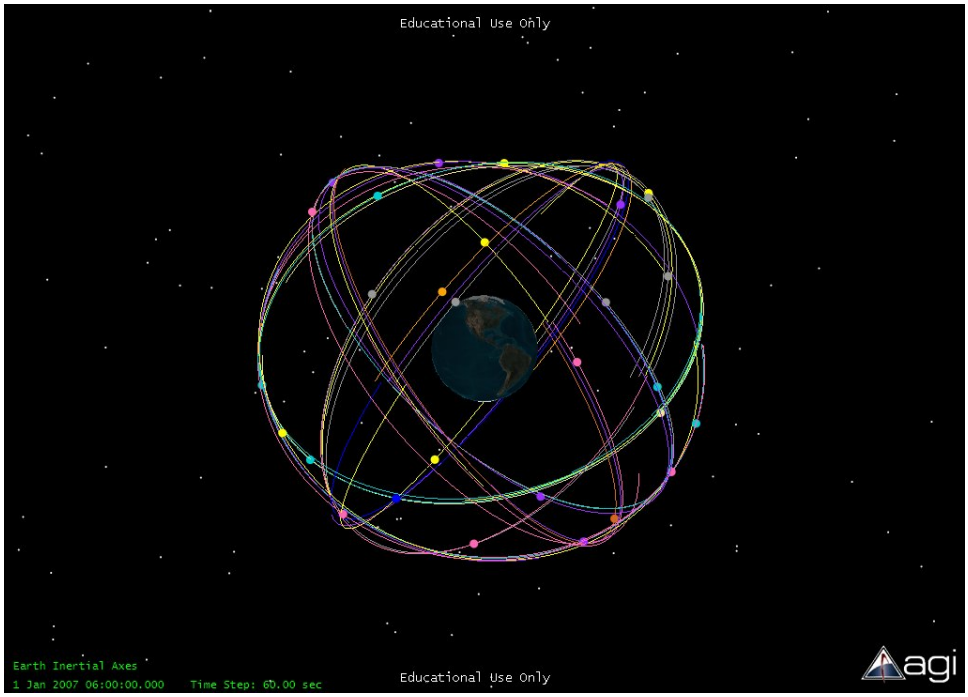

Figure 2.1: GPS Constellation, 1 January 2007

the current time. The receiver uses this information to calculate it's position based on knowledge of the satellite's position and the time elapsed from transmission until the message is received. A given ephemeris file is valid for four hours and overlaps with the file before it by two hours. Each of the ephemeris files provides essentially the average osculating orbital elements over the time period for which it is valid. Currently all the ephemeris files for a satellite during a given day are uploaded once a day. The ephemeris files for all of the satellites are available through the National Geodetic Survey (NGS) [NGS, Aug 2007] and are maintained by the International GNSS Service (IGS). IGS coordinates the tracking of Global Navigation Satellite System (GNSS) satellites using a global network of antennas and receivers. This information is used to calculate GPS final (precise) orbits. The final orbits are published weekly and are available on the web. The final GPS satellite orbit data, published approximately 13 days after a given week is over, has an accuracy of ≤ 5 cm. Broadcast ephemeris data for GPS satellites has an accuracy of $\sim$160 cm [IGS, 2005].

Each satellite has a unique Pseudo Random Number (PRN). This number is the method that receivers use to differentiate between satellites. There are 32 possible PRNs.

When the constellation was initially created the PRNs corresponded to the Satellite Vehicle Number (SVN), but now that satellites have been retired and new satellites have been launched, PRNs do not necessarily correspond to satellite numbers. The PRNs are provided in the GPS precise orbit data and the broadcast ephemeris data to identify satellites. Throughout this thesis the PRN numbers are used for reference rather than satellite numbers. Table 2.1 shows all of the satellites that were in continuous operation, without any station keeping maneuvers, during 2007. These were chosen to allow six months of final orbit data for analysis followed by six months of orbit data for comparison of predicted location values against actual positions. The precise orbit data provided by IGS gives the location of each satellite at 15-minute time intervals.

2.2 Orbit Modeling

Current orbit modeling typically uses numerical integration. This method can be time consuming. To predict the future location of a satellite one must determine the orbit at every point leading up to the point of interest. In the past, this calculation could take almost as long as for the satellite to move through the orbit to the point of interest. With the advent of more powerful computers, the relative computational intensity and time to do a numerical integration has decreased. Even with powerful computers, the cumulative computational requirements for predicting and tracking several objects simultaneously remain large.

Currently the US Space Surveillance Network is tracking over 12,000 objects in Earth orbit [NASA, 2008]. Figure 2.2 below shows the increased number of Earth orbiting objects. The sharp increase in debris during 2007 is a result of the destruction of Fengyun-1C on 11 January 2007 by the People's Republic of China as a test of an anti-satellite system. All objects orbiting Earth must be tracked to reduce the risk of hypervelocity impacts. In 1983, a small paint chip damaged the windshield on the Challenger shuttle, thus demonstrating the damaging power of small items in space [OTA, 1990]. Militarily, another reason to track satellites is for situational awareness, especially in the case of spy satellites flying over sensitive areas.

Table 2.1: GPS Satellites in Operation, 2007

PRN	Plane	Launch Date	SSC#
25	A5	23 Feb 1992	21890
26	F2	07 Jul 1992	22014
27	A4	09 Sep 1992	22108
01	A6	22 Nov 1992	22231
09	A1	26 Jun 1993	22700
05	B4	30 Aug 1993	22779
04	D4	26 Oct 1993	22877
06	C1	11 Mar 1994	23027
03	C2	28 Mar 1996	23833
30	B2	12 Sep 1996	24320
13	F3	23 Jul 1997	24876
08	A3	06 Nov 1997	25030
11	D2	07 Oct 1999	25933
20	E1	11 May 2000	26360
28	B3	16 Jul 2000	26407
14	F1	10 Nov 2000	26605
18	E4	30 Jan 2001	26690
16	B1	29 Jan 2003	27663
21	D3	31 Mar 2003	27704
22	E2	21 Dec 2003	28129
19	C3	20 Mar 2004	28190
23	F4	23 Jun 2004	28361
02	D1	06 Nov 2004	28474
17	C4	26 Sep 2005	28874
31	A2	25 Sep 2006	29486
12	B5	17 Nov 2006	29601

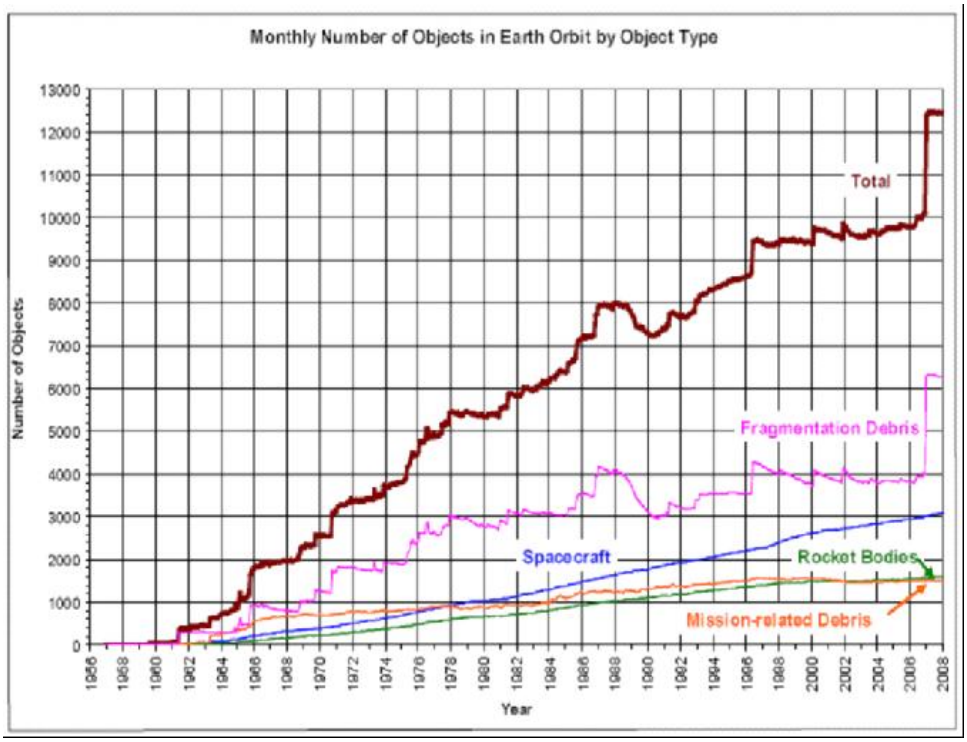

Figure 2.2: Earth Orbiting Objects, January 2008

During a geomagnetic storm, low earth orbiting satellite drag rapidly increases due to thermospheric heating [*Campbell*, 2003]. This change can significantly alter a satellite's orbit to the point where automated tracking software loses them. Increased drag causes a satellite to lose altitude, which results in a higher velocity. This was the case with the geomagnetic storm on 13-14 March 1989. Figure 2.3 below shows the increased number of lost satellites following the storm relative to the geomagnetic index. It took the North American Defense Command (NORAD) several days to reacquire the thousands of objects that were lost. During the Halloween space weather storms of 2003, Air Force Space Command used satellite drag models to correct for orbital changes. These models were based on the advanced warning geomagnetic and solar activities indices [NOAA, 2004] .

2.3 Satellite Dynamics

The motion of satellites, for the application of KAM theory, must be considered in the Earth Centered Earth Fixed (ECEF) rotating reference frame. It is in this frame

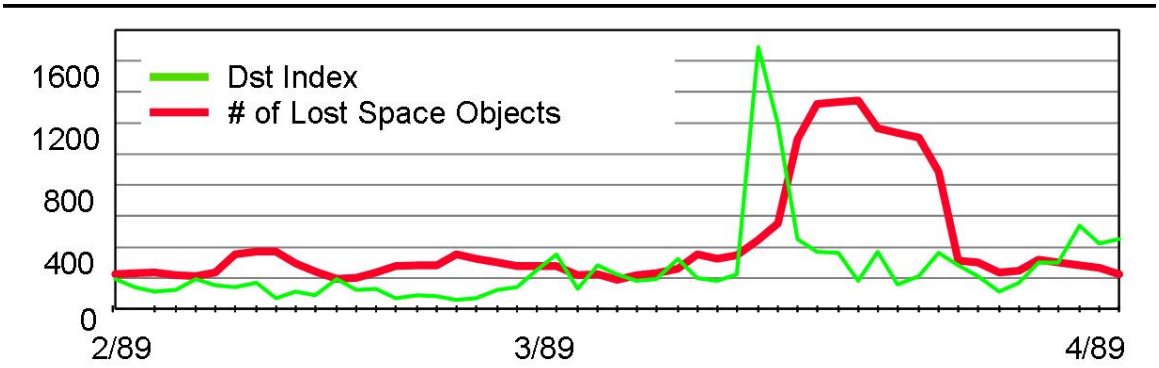

Figure 2.3: March 1989 Geomagnetic Storm

that the Earth's geopotential gravity field is constant with only small smooth variations as an object moves around the Earth. The ECEF frame is a Cartesian coordinate system where the axes are defined with the x and y axes in the plane of the equator and the x axis points through the prime meridian. The z axis points out of the North pole to complete the right handed coordinate system. The inertial velocity components of a satellite may be written in the ECEF reference frame as shown by Equation 2.1. In this equation the inertial velocity components have been converted to the rotating reference frame. The positions in the ECEF frame are given by x, y, z and the inertial velocities are given by ẋ, ẏ, ż. $\omega_{\oplus}$ is the angular velocity of the Earth.

$$v = \left\{ \begin{array}{c} \dot{x} - \omega_{\oplus} y \\ \dot{y} + \omega_{\oplus} x \\ \dot{z} \end{array} \right\} \tag{2.1}$$

The kinetic energy of the satellite per unit mass is given by Equation 2.2

$$T = \frac{1}{2}((\dot{x} - \omega_{\oplus} y)^2 + (\dot{y} + \omega_{\oplus} x)^2 + \dot{z}^2) \tag{2.2}$$

The momenta p_i are defined as $p_i = \delta T / \delta \dot{q}_i$ where q_i are the generalized coordinates and $\dot{q}_i$ are the time derivatives of these coordinates. In this formulation the $\dot{q}_i$s are given by the components of the velocity in Equation 2.1. Equations 2.3-2.5 give the momenta

for the earth orbiting satellite in the ECEF frame.

$$p_x = \dot{x} - \omega_{\oplus} y \tag{2.3}$$

$$p_y = \dot{y} + \omega_{\oplus} x \tag{2.4}$$

$$p_z = \dot{z} \tag{2.5}$$

The potential energy per unit mass of the satellite is given by Equation 2.6. This is the expansion of the geopotential in spherical harmonics [*Wiesel*, 2003].

$$V = -\frac{\mu}{r} \sum_{n=1}^{\infty} \sum_{m=1}^{n} (\frac{r}{R_{\oplus}})^{-n} P_n^m(sin\delta) * (C_{nm} cosm\lambda + S_{nm} sinm\lambda) \tag{2.6}$$

In this equation, μ is the Earth's gravitational parameter and $R_{\oplus}$ is the radius of the earth. The functions P_n^m are the associated Legendre polynomials. C_{nm} and S_{nm} are coefficients that specify the gravitational field. Several models are available with these values. For the analysis in this thesis, the harmonic terms for the geomagnetic field of the earth are taken from National Aeronautics and Space Administration's (NASA)'s Earth Gravitational Model (EGM) 96. For the numerical integration, EGM 96 was used to order and degree n,m < 20. Figure 2.4 [NASA, 1998] depicts the EGM 96. The full EGM 96 is available in tabular format on the web [NASA, 1998]. In Equation 2.6 the radius r, geocentric latitude δ and east longitude λ are found using the following equations:

$$r = \sqrt{x^2 + y^2 + z^2}$$

$$sin\delta = \frac{z}{\sqrt{x^2 + y^2}}$$

$$tan\lambda = \frac{y}{z}$$

The Hamiltonian is formed using $H = \sum p\dot{q} - T + V$ which is equivalent to

$$H = \frac{1}{2}(p_x^2 + p_y^2 + p_z^2) + \omega(yp_x - xp_y) - \frac{\mu}{r} \sum_{n=1}^{\infty} \sum_{m=1}^{n} (\frac{r}{R_{\oplus}})^{-n} P_n^m(sin\delta) * (C_{nm} cosm\lambda + S_{nm} sinm\lambda) \tag{2.7}$$

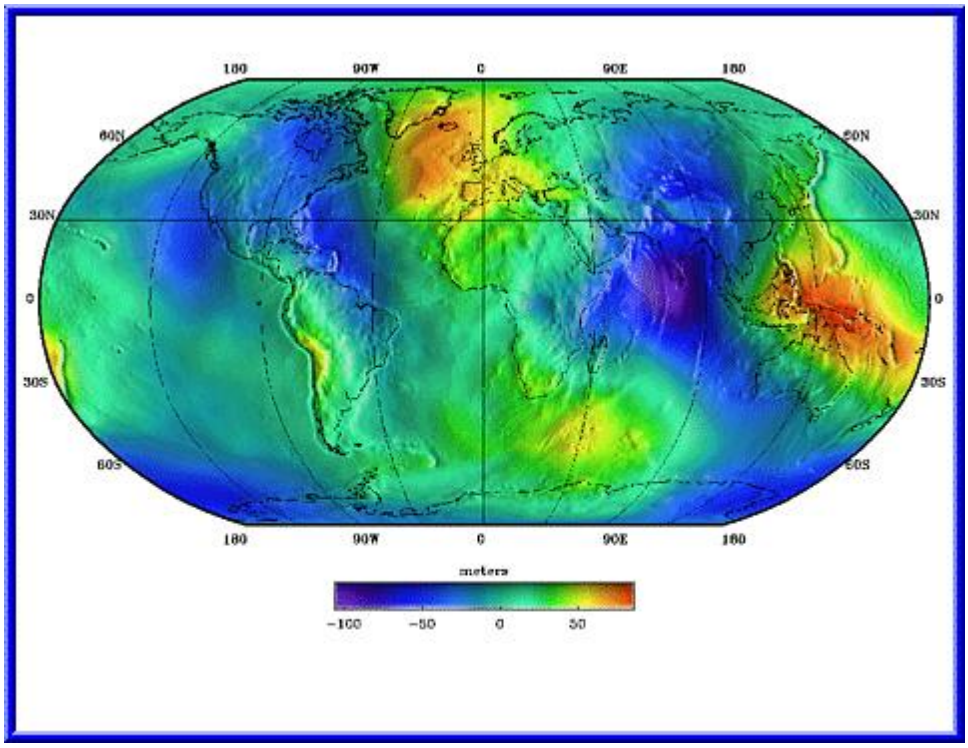

Figure 2.4: Earth Gravitational Model 96 Geoid

The Hamiltonian is independent of time, which means that it must be a constant of the motion.

2.4 Kolomogorov, Arnold, Moser Tori

Kolmogorov [*Kolmogorov*, 1954], Arnold[*Arnold*, 1963], and Moser [*Moser*, 1962] developed theories that together form the KAM theorem. The necessary conditions for the KAM theorem to apply are that there are only small, smooth perturbations to the Hamiltonian. A Hamiltonian that follows KAM theory can be represented by a torus with discrete frequencies the number of which is equivalent to the number of coordinates of the system. An N-dimensional system is represented in 2N-dimensional phase space. Figure 2.5 depicts a three dimensional torus. A trajectory lying on a torus will have quasiperiodic motion and remain on the torus in the future. Kolmogorov's and Arnold's works were published in Russian and therefore unavailable for review. Several other authors have, however, provided summaries of their theories.

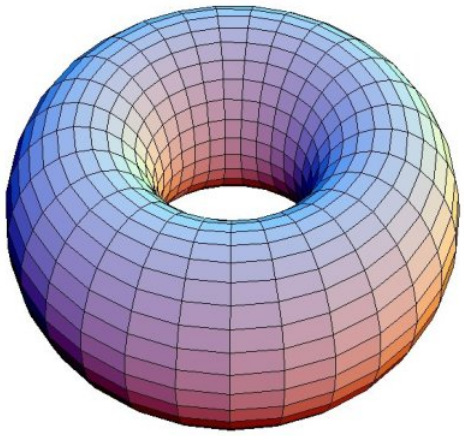

Figure 2.5: Representation of a 3 Dimensional Torus

Kolmogorov stated that for a nearly integrable Hamiltonian in phase space $M := V \times \mathbb{T}^d$, the Hamiltonian function is given by 2.8. In the phase space definition d is the number of dimensions.

$$H_\varepsilon(I, \varphi) := h(I) + \varepsilon f(I, \varphi) \tag{2.8}$$

where h and f are real-analytic functions, ε is a small real parameter, and variables I and φ are sympletic action-angle variables.

Kolmogorov's theorem states that: *In any neighborhood of any torus* $I_0 \times \mathbb{T}_d \subset M$ *such that*

$$\det h''(I_0) := \det \left(\frac{\delta^2 h}{\delta I_i \delta I_j}(I_0) \right)_{i,j=1,\ldots,d} \neq 0, \tag{2.9}$$

there exists a positive measure set of phase points belonging to analytic KAM tori for H_ε*, provided* ε *is small enough.* [*Celletti*, 2006] The measure is the 2d-dimensional Liouville measure in phase space.

In a Hamiltonian where h, as given by Kolmogorov, does not depend on all of the action angles, the system is properly degenerate. In this case, KAM tori cannot be identified (and may not exist) without additional information about the perturbation, f, of the Hamiltonian. Arnold focused on this special case by attempting to apply his theorem to the planetary many body problem. Arnold's formulation begins similar to Kolmogorov's, with M designating the phase space and the Hamiltonian given by $H_\varepsilon(I, \varphi, p, q) := h(I) + \varepsilon f(I, \varphi, p, q)$. The average power of f over the "fast angles" φ is

given by

$$\bar{f}(I,p,q) := \int_{\mathbb{T}^d} f(I,\varphi,p,q)\frac{d\varphi}{(2\pi)^d} \tag{2.10}$$

Arnold's theorem states: *Assume that* $\bar{f}$ *is of the form*

$$\bar{f} = f_0(I) + \sum_{j=1}^{m}\Omega_j(I)J_j + \frac{1}{2}A(I)J\cdot J + o_4; J_j := \frac{p_j^2+q_j^2}{2}, \tag{2.11}$$

where A *is a symmetric (m x m)-matrix and* $\lim_{(p,q)\to 0}|o_4|/|(p,q)|^4 = 0$. *Assume, also, that* $I_0 \in V$ *is such that*

$$\det h''(I_0) \neq 0, \tag{2.12}$$

$$\sum_{j=1}^{m}\Omega_j(I_0)k_j \neq 0, \qquad \forall k \in \mathbb{Z}^m \text{ with} 0 < \sum_{j=1}^{m}|k_j| \leq 6, \tag{2.13}$$

$$\det A(I_0) \neq 0. \tag{2.14}$$

Then, in any neighborhood of $I_0 \times \mathbb{T}^d \times (0,0) \subset M$ *there exists a positive measure set of phase points belonging to analytic KAM tori for* H_ε, *provided* ε *is small enough.* [*Celletti*, 2006]

Kolmogorov's theorem focused on analytic Hamiltonians with near integrable differential equations. For these he showed the existence of quasiperiodic solutions. Moser formulated his problem in a geometric fashion in an attempt to verify Kolmogorov's theorem. Moser defines the mapping (including perturbation of a twist mapping), assuming F and G are small with period 2π for θ, as

$$\theta_1 = \theta + \alpha(r) + F(r,\theta) \tag{2.15}$$

$$r_1 = r + G(r,\theta) \tag{2.16}$$

The second assumption is that every closed curve which is near a circle (r =const) has $r = f(\theta) = f(\theta+2\pi)$ and with $f'(\theta)$ small, the closed curve and it's image curve intersect.

Moser's theorem states: *For a given $\varepsilon > 0$ and a given integer $s \geq 1$ the mapping has a closed invariant curve*

$$\theta = \theta' + p(\theta') \tag{2.17}$$

$$r = r_0 + q(\theta') \tag{2.18}$$

where the functions p,q are functions of period 2π with s continuous derivatives satisfying

$$|p|_s + |q|_s < \varepsilon \tag{2.19}$$

under the following hypotheses: Assume for the mapping that every closed curve near a circle and its image curve intersect. Assume further $b - a \geq 1$ and

$$c_0^{-1} \leq \frac{d\alpha(r)}{dr} \leq c_0 \tag{2.20}$$

with some constant $c_0 > 1$. Finally construct a positive number $\delta_0 = \delta_0(\varepsilon, s, (c_0))$ and an integer l=l(s) with which it is required that F,G have continuous derivatives up to order l and satisfy the inequalities

$$|F|_0 + |G|_0 < \delta_0 \tag{2.21}$$

$$|\alpha|_l + |F|_l + |G|_l < c_0 \tag{2.22}$$

Moreover, the mapping induced on the curve is given by

$$\theta_1' = \theta' + \alpha(r_0) \tag{2.23}$$

[*Moser*, 1962]

2.5 Earth-Satellite KAM

The basis frequencies of the tori in the ECEF frame are given in Equations 2.24 - 2.26 [*Wiesel*, 2007]. All of these fundamental frequencies can be approximated in terms of the classical orbital elements and are listed in order of size with ω_1 being the largest

and ω_3 being the smallest of the frequencies. The first frequency is the anomalistic frequency.

$$\omega_1 = \sqrt{\frac{\mu}{a^3}} \left\{ 1 - \frac{3J_2 R_\oplus^2}{2a^2(1-e^2)^{3/2}} (\frac{3}{2}\sin^2 i - 1) \right\} \tag{2.24}$$

The second frequency is a combination of the earth's rotational frequency and the nodal regression rate.

$$\omega_2 = \omega_\oplus + \frac{3\sqrt{\mu} J_2 R_\oplus^2}{2a^{7/2}(1-e^2)^2} \cos^2 i \tag{2.25}$$

The final frequency is the apsidial regression rate.

$$\omega_3 = \frac{3\sqrt{(\mu)} J_2 R_\oplus^2}{2a^{7/2}(1-e^2)^2} (\frac{5}{2}\sin^2 i - 2) \tag{2.26}$$

Where $R_\oplus$ is the radius of the Earth, μ is the Earth gravitational parameter, J_2 is the J_2 term of the geopotential, $\omega_\oplus$ is the Earth rotation frequency, e is the orbit eccentricity, a is the orbit semi-major axis, and i is the orbit inclination. All the frequency equations are independent of the right ascension of a satellite.

The motion of the satellite in the z axis of the ECEF coordinate frame is independent of the Earth's rotation and is therefore given by multiples of the mean motion. The mean motion of a satellite is given by Equation 2.27.

$$n = \sqrt{\frac{\mu}{a^3}} \tag{2.27}$$

The actual frequencies are identified by doing FFTs on the satellite position in each coordinate of the ECEF frame. The equation for identifying the position of a satellite is based on the frequencies identified and is given by Equation 2.28. C and S are the Fourier series coefficients.

$$\mathbf{X} = \sum_{ijk}^{\infty} C_{ijk} cos((i\omega_1 + j\omega_2 + k\omega_3)t) + S_{ijk} sin((i\omega_1 + j\omega_2 + k\omega_3)t) \tag{2.28}$$

Where $\mathbf{X}$ is the state matrix of a satellite at time, t, given by $\mathbf{X} = \{\text{x y z}\}^T$

The period for a satellite to travel the entire torus is based on the time for the smallest frequency to traverse a circle. Equation 2.29 gives the period of the torus.

$$T = \frac{2\pi}{\omega_3} \tag{2.29}$$

2.6 Laskar Frequency Algorithm

Laskar [*Laskar*, 1999] [*Laskar*, 2003] provides the the algorithm for an accelerated Fourier transform to identify the frequencies of a quasiperiodic function more precisely than with a simple FFT. For a quasiperiodic function evaluated over the interval $[-\tau : \tau]$ an ordinary FFT assumes the function is periodic with a period of 2τ, which is not typically the case. Laskar's Numerical Algorithm of the Fundamental Frequency (NAFF) determines the frequencies without this limitation. For an ordinary FFT the accuracy of the solution for the frequencies is proportional to $1/\tau$. The NAFF accuracy is proportional to $1/\tau^2$. This is further refined using a Hanning weighting to produce the frequencies of a KAM solution with accuracies proportional to $1/\tau^4$. The Hanning weighting function is given by Equation 2.30.

$$\chi(t/\tau) = 1 + \cos\left(\frac{\pi t}{\tau}\right) \tag{2.30}$$

The NAFF given by Laskar is to find the maximum amplitudes in Equation 2.31 through an iterative method.

$$\phi(\omega) = \frac{1}{2\tau}\int_{-\tau}^{\tau} f(t)e^{-i\omega t}\chi(t/\tau)dt \tag{2.31}$$

Using approximate frequencies, identified through independent numerical integration and an ordinary FFT, the peak values in Equation 2.31 can be converged upon in a moderate number of iterations.

2.7 KAM Theory Applied

McGill and Binney show that most orbits are approximately quasiperiodic and they can be represented by a torus in phase-space. A method for doing linear least squares fitting to identify the orbital torus is discussed [*McGill and Binney*, 1990]. A toy Hamiltonian, H_0, is represented by an analytic tori. The target tori is the Hamiltonian of interest, H_ε. Based on perturbation theory, the distortion of the toy tori into the target tori uses a generating function and a canonical transformation. The technique for identifying the orbital torus requires that a toy Hamiltonian is available that can be mapped to the target torus.

Beginning with Arnold's attempt to apply KAM theory to the restricted three-body problem, astronomers have worked to apply the theory to celestial mechanics. Arnold started by posing the question, "Do there exist, in the n-body problem, a set of initial conditions having positive measure such that, if the initial position and velocities of the bodies belong to this set, then the distances of the bodies from each other will remain perpetually bounded?" [*Celletti*, 2006] This is true in the special case of the restricted, planar, three-body problem (RPC3BP). Initial general attempts to apply KAM theory to the Solar System provided poor results because the parameter ε, the mass ratio, needed to be small. Celletti and others have completed several applications of the KAM theory. In the context of the RPC3BP the Sun-Jupiter-Ceres [*Celletti*, 1998], Sun-Jupiter-Saturn [*Laskar*, 2003], and the Sun-Jupiter-Victoria [*Celletti et al.*, 2004] [*Celletti and Chierchia*, 2005] systems were analyzed. The numerical studies completed on the Sun-Jupiter-Victoria truncated model show results very close to those obtained using the complete perturbation function [*Celletti*, 2006]. Moser's theorem provides an estimate for the mass ratio of two bodies of less than 10^{-50}; which is the desired value for the two primary bodies. In the Sun-Jupiter case ε is only 10^{-3}, but with the use of computers, it is possible to obtain a result close to reality [*Celletti et al.*, 2004].

III. Method

This work is based on the KAM theorem. It is applied to precise GPS data to verify the existence of discrete frequencies. FFTs were used to identify the frequencies.

3.1 Data Gathering

GPS final precise files were downloaded from the NASA server [IGS, 2007] using a shell script on an Ubuntu server version 7.10. The broadcast ephemeris data was downloaded in a similar manner. GPS data is provided based on the GPS week number. A calender is available on the web for easy identification of the GPS weeks relative to a standard calendar [NGS, 2007]. All GPS data is given with positions in the ECEF frame and time given by GPS time.

GPS final orbit files are in sp3 format [*Hilla*, 2007]. The first 22 lines of code contain comments and the remainder of the file is in the form seen in Figure 3.1.

```
*  2007  1  1  0  0  0.00000000
PG01 -13611.975739  13373.070070 -18238.588923     98.711251 11 10 10 172
PG02  22725.438511 -13205.010044   2507.995439     69.157928 13 10 13 201
PG03 -11737.674285  17133.489073  16173.165844    518.252288 11 12 14 191
PG04  24622.216474  -3778.903342  -9780.051158    350.987566  9 14 10 184
PG05  -5181.092119 -18497.996064 -18635.867925    454.598118 11  8  9 164
PG06  -7908.679569 -22552.665443  11714.643028    481.111422  9 10  8 158
PG07 -11359.909279 -18490.081281  15593.644644    459.082775 10 10  8 167
PG08  19893.750080   4060.771479  17228.643849    -91.699044 13 13 11 201
PG09   8353.931698 -18401.652165 -17688.405375     51.181664 14  9  8 215
PG10  13807.306063  -7970.868750  21196.376445     95.217252 13 11 10 182
PG11  -4399.422547  19829.470893 -17072.576710      1.823003 12  9 11 167
PG12  -4081.740525 -18394.893078 -18630.298051    -72.851199 13  8 11 187
PG13   7105.535401  21288.037015  14151.698360    128.126912 12  9  9 164
PG14 -14152.692634  -4465.479069 -21949.037533      3.325293 13 13  8 186
PG15 -21023.797955  -3760.996784  15932.503266    352.229108 12 12 14 216
PG16 -17910.100936   3961.317154  19331.708991    107.692958  9 13 10 178
PG17  14850.925170   4411.000084 -21590.672150     84.177813 11 14 11 170
PG18 -19503.682845 -17875.147550   3986.097722   -253.769575 10 10 12 163
PG19  -8614.175325  23914.476412   7634.534549     51.286456  9  4 10 174
PG20   6873.011101  19115.540361 -17223.146829    -21.882901 12  9  9 181
PG21 -15090.628398  -9258.926179  20042.260790     66.977665 10 10  7 179
PG22 -23121.567838 -10413.946351  -7770.365432    159.323004 11 12 11 174
PG23    729.451999  26173.678549   3712.159995    144.461494 12  8  8 150
PG24  17791.343107 -16393.380090  11541.401896     64.771961 11  8 10 183
PG25 -23840.456545  10870.015685  -4452.652542    394.051853 12 11 10 195
PG26   8025.278328 -23335.770084   8593.809269    -73.036644  9  8  9 162
PG27  11963.093357  10714.109648  21829.746143     86.166055 14 13 11 200
PG28  22980.656636  13117.231243  -3626.921139     10.642863 12 13 12 186
PG29  10156.502411 -20570.051777  13313.071492    292.094773 20 18 17 155
PG30 -15406.742034 -18903.114013 -10960.438838     21.199089  8 10 10 155
PG31 -24705.789337   6846.874604  -6834.813768    -11.394328 12  9 10 195
```

Figure 3.1: Final Orbit Data File, sp3 format

In a final orbit file the epoch identification lines have an asterisk in the first column. The remaining entries on this line are as follows: year, month, day of month, hour, minutes, seconds. The position and clock record for satellites are on lines beginning with PG. Columns three and four are the PRN identifying a given satellite. The remaining entries are in order: the x, y, and z coordinates in km, the clock given in microseconds, the standard deviations for each of the components x, y, z, and the clock. The analysis completed for this thesis used only the epoch header information, the PRN and coordinates of each satellite.

GPS broadcast ephemeris files are in RINEX format [*Gurtner*, 2002]. The first 3 lines of code contain comments and the remainder of the file is in the form seen in Figure 3.2.

```
 1 07  1  2  0  0  0.0 0.989157706499E-04 0.238742359215E-11 0.000000000000E+00
    0.180000000000E+02-0.695000000000E+02 0.402766776853E-08-0.361491914275E+00
   -0.359117984772E-05 0.671335251536E-02 0.888481736183E-06 0.515372728157E+04
    0.172800000000E+06-0.465661287308E-07-0.143951461866E+01 0.745058059692E-08
    0.989187963695E+00 0.376343750000E+03-0.176594710576E+01-0.823320008869E-08
   -0.242152943785E-09 0.000000000000E+00 0.140800000000E+04 0.000000000000E+00
    0.280000000000E+01 0.000000000000E+00-0.325962901115E-08 0.180000000000E+02
    0.171150000000E+06 0.000000000000E+00 0.000000000000E+00 0.000000000000E+00
 2 07  1  2  0  0  0.0 0.694547779858E-04 0.352429196937E-11 0.000000000000E+00
    0.120000000000E+03 0.715312500000E+02 0.468590947265E-08 0.858699980781E+00
    0.355765223503E-05 0.885681062937E-02 0.111702829599E-04 0.515376655006E+04
    0.172800000000E+06 0.167638063431E-07 0.270030433369E+01-0.210478901863E-06
    0.948047556180E+00 0.158312500000E+03 0.218769731207E+01-0.790532928870E-08
   -0.495734935064E-09 0.100000000000E+01 0.140800000000E+04 0.000000000000E+00
    0.280000000000E+01 0.000000000000E+00-0.172294676304E-07 0.376000000000E+03
    0.165618000000E+06 0.400000000000E+01 0.000000000000E+00 0.000000000000E+00
```

Figure 3.2: Broadcast Ephemeris, RINEX format

The file is in groups of eight lines of data per satellite. The first line contains the PRN number of the satellite, the year, month, day, hours, minutes, and second (of the epoch for which the parameters apply), clock offset, rate, and acceleration. The second line contains the age of the ephemeris entry, radius correction, correction to the mean motion, and mean anomaly. The third line contains a correction to the argument of latitude, eccentricity, a second argument of latitude correction and the square root of the semi-major axis. The fourth line has the time of ephemeris, correction to the inclination, longitude of the ascending node, and a second correction to the inclination. The fifth line contains the inclination, a radius correction, argument of perigee, and the time derivative of the longitude of the ascending node. The first parameter in the sixth line is the time derivative of the inclination. None of the remaining values on the sixth

line or any values in lines seven and eight are needed to calculate the satellite dynamics at a point in time.

The final orbit files and the broadcast ephemeris files were consolidated into their own respective data files, eliminating the comments to reduce processing time. GPS satellites are designated by PRN number. Matlab code was written to step through the final orbit file, extracting the x, y, and z positions and times for the input set of satellites identified by their PRN. A similar code was written to step through the broadcast ephemeris file to gather the values required to calculate the satellite velocity at a given time. A discussion of the method to calculate the velocities is in Section 3.3. The complete code is in Appendix C for reference.

3.2 Position Frequencies

An estimate of the mass ratio ε for a GPS satellite and Earth gives a value of 3.348e-22. This uses an approximate GPS satellite weight of 2e3 kg [AFSPC, 2007] and the mass of the Earth as 5.9742e24 kg. This calculation does not give as small a value as desired and discussed in Section 2.7, however, with the use of computers to identify the KAM solution, it remains possible.

An initial estimate of the expected frequencies was completed using the equations given in Section 2.5.

In the ECEF reference frame each position coordinate was analyzed independently. A FFT was completed on the x, y, and z position vectors. With L defined as the length of the FFT vector, ϕ, and a nyquist frequency, $\eta = 0.5$, the frequency is calculated over the interval [1:0.5L] as $\frac{\phi}{L/2}\eta$. The power corresponding to these frequencies is given by $|\phi|^2$. The power and frequency are plotted. A log scale is used for the power axis and the frequencies are in orbits/15minutes because that is the time scale of data in the final orbit file.

3.3 Computing Velocities from Broadcast Ephemeris Data

Building on the calculations in the GPS ICD-200 [*ARINC Research Corporation*, April 2000] receiver interface, it is possible to calculate the satellite velocity in the ECEF

coordinate frame. The details of these calculations are provided by Remondi [*Remondi*, 2004a], including an example C code available on the web [*Remondi*, 2004b]. The satellite velocities were calculated using two methods to validate the code. This code was converted into Matlab code and validated with the sample file given by Remondi. The code is included in the Appendix.

3.4 Sidereal Time

Until this point all calculations have been completed in the ECEF reference frame and times have been converted to Julian dates for compact representation of the date and time. In order to convert the ECEF values into the Earth Centered Inertial (ECI) reference frame, GPS time must be converted to UTC time. Section 2.1 describes the time difference between these systems. In Julian date format, this is equivalent to adding .0016 to the GPS Julian date to obtain a UTC Julian date. The United States Naval Observatory (USNO) provides the formulas needed to calculate the Greenwich Apparent Sidereal Time angle (GAST) based on a Julian date [USNO, 2008]. This angle is the rotation between the ECEF frame and the ECI frame. This method will give results on the order of 10^{-7} radians. Precise GPS data has an accuracy on the order of 10^{-9} radians. Calculating GPS satellite dynamics to this level of accuracy in the ECI frame based on precise data would require use of the Multiyear Interactive Computer Almanac [USNO, 2006].

3.5 Integrated Orbit Frequency Set

A hypothetical GPS satellite data point was developed using basic satellite dynamics. This was developed using an ideal satellite with $i = 55°$, $e = 0$, and $a = 26{,}560$ km. The ECEF and ECI reference frames were assumed to be aligned at the moment of interest with the satellite on the x axis at the ascending node.

For a satellite in a circular orbit the velocity tangent to the orbit is given by Equation 3.1

$$v = \sqrt{\frac{\mu}{a}} \tag{3.1}$$

Values were converted to canonical units for the analysis. Canonical units of Distance Unit (DU) and Time Units (TU) are defined where 1 DU = 6378.135 km (radius of the earth) and 1 TU = 13.44686457 min. The GPS satellite position and velocity have been input into a numerical integration based orbit propagator. The orbit data generated was fit with a FFT to identify the frequencies.

3.6 Laskar Frequency Fitting

The integrated orbit created and frequencies identified in Section 3.5 are refined using the Laskar frequency fitting algorithm to get better resolution. In practice, this can be a relatively time consuming process to achieve convergence; therefore, it is important to have approximate frequencies to several significant digits as identified through an ordinary FFT.

IV. Results and Discussion

This chapter shows there are discrete frequencies for most GPS satellites. Some of the older satellites are in semi-stable orbits. Satellites in each orbital plane, for the most part, have the same orbital frequencies.

4.1 Frequency Estimates

Initial frequency estimates were calculated using the frequency equations in Section 2.5 based on the orbital elements. These estimates used the following values for Earth constants: $\mu_\oplus = 3.986012e5$ km^3/s^2, $R_\oplus = 6378.145$ km, $J_2 = 0.00182$, and $\omega_\oplus = 7.292115856e-5$ rad/s. GPS orbit values of $e = 0.0032$, $a = 26560.62369$ km, and $i = 55^\circ$ were used in the calculations. The estimated frequencies are therefore:

$$\begin{aligned}\omega_1 &= 1.4585e-4 \ \frac{\text{rad}}{\text{sec}} \\ \omega_2 &= 7.2929e-5 \ \frac{\text{rad}}{\text{sec}} \\ \omega_3 &= -4.4020e-9 \ \frac{\text{rad}}{\text{sec}}\end{aligned}$$

For comparison with the results in later sections, these frequencies are also equivalent to:

$$\begin{aligned}\omega_1 &= 2.0892e-2 \ \frac{\text{orbits}}{15\text{min}} = 1.1767e-1 \ \frac{\text{rad}}{\text{TU}} \\ \omega_2 &= 1.0446e-2 \ \frac{\text{orbits}}{15\text{min}} = 5.8840e-2 \ \frac{\text{rad}}{\text{TU}} \\ \omega_3 &= -6.3054e-7 \ \frac{\text{orbits}}{15\text{min}} = -3.5516e-6 \ \frac{\text{rad}}{\text{TU}}\end{aligned}$$

The mean motion for a GPS satellite using Equation 2.27 and the ideal value of $a = 26560$ km gives $n = 1.458569725$e-4 rad/sec. This is equivalent to ω_1 (with the exception of the J_2 which is small). It is expected that the frequencies in the z coordinate will therefore be multiples of ω_1. Calculation of the period of the torus using Equation 2.29 gives a value on the order of 19 years for a GPS satellite to traverse the entire KAM torus of it's orbit.

4.2 Position Frequencies

FFTs were completed independently on each of the ECEF coordinate positions. Of the 26 satellites analyzed that were in operation during 2007, 25 had stable frequency mappings. The remaining satellite shows a semi-stable frequency map. Because the frequencies can be written in terms of the orbital elements, independent of right ascensions, as shown in Section 2.5, it is expected that all satellites will have identical frequency maps.

The graphs in Figure 4.1 show the x, y, and z position frequencies of a satellite in the A orbital plane.

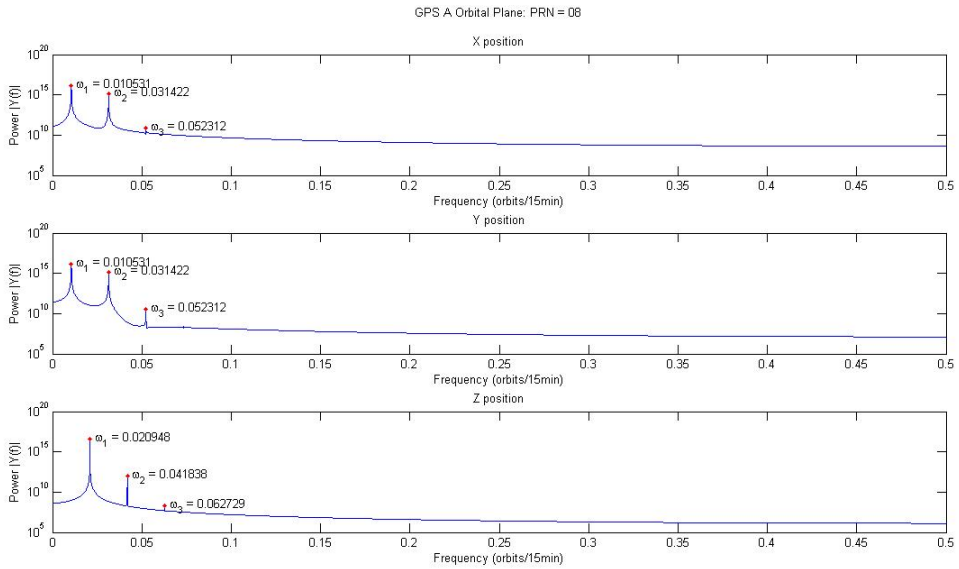

Figure 4.1: Frequencies of positions for PRN 08 located in the A orbital plane

Peak analysis of this plot identifies the frequencies for each axis as shown in Table 4.1. The x and y coordinates have the same frequency values to the order shown in this analysis. This is likely due to the symmetry of the orbit relative to these axes.

All the orbital planes have very close or identical orbital frequencies. This is expected since the orbits in each of the planes are identical, with the exception of the RAAN. As discussed earlier, the frequencies do not depend on the RAAN. The graphs below show the results in each of the remaining five orbital planes. The frequency graphs for all of the satellites in operation during 2007 are in the Appendix for reference.

Table 4.1: Precise Satellite Orbit Frequencies, PRN 08

Coordinate	Frequency $\frac{\text{orbits}}{15\text{min}}$	Identity
x	0.010531	ω_2
y	0.010531	ω_2
z	0.020948	ω_1
x	0.031422	$\omega_1 + \omega_2$
y	0.031422	$\omega_1 + \omega_2$
z	0.041838	$2\omega_1$
x	0.052312	$\omega_2 + 2\omega_1$
y	0.052312	$\omega_2 + 2\omega_1$
z	0.062729	$3\omega_1$

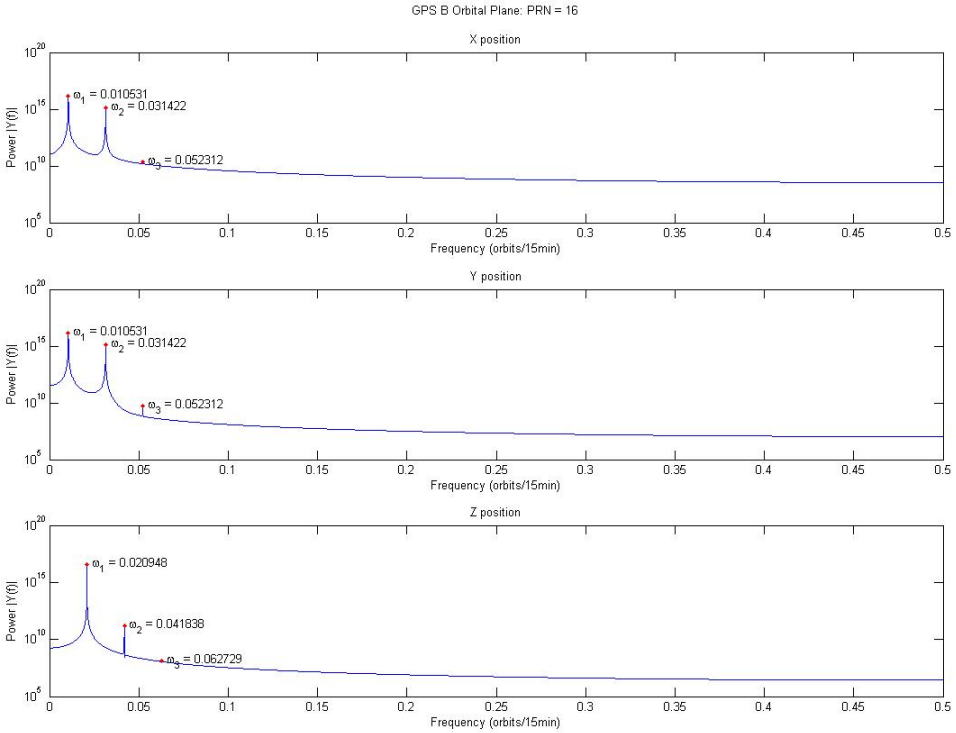

Figure 4.2: Frequencies of positions for PRN 16 located in the B orbital plane

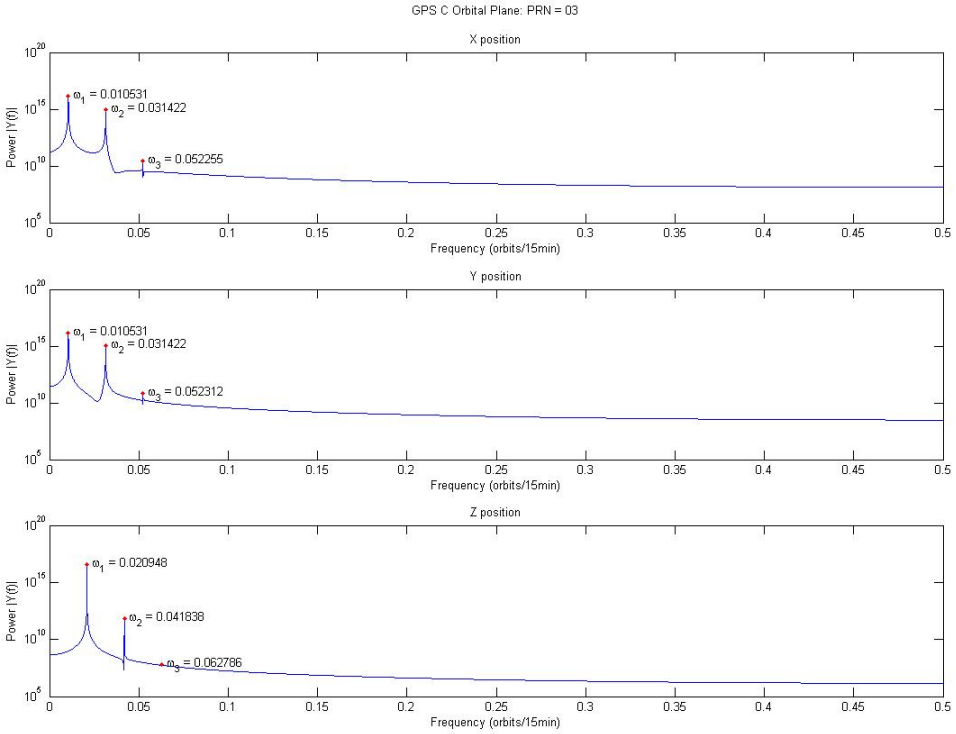

Figure 4.3: Frequencies of positions for PRN 03 located in the C orbital plane

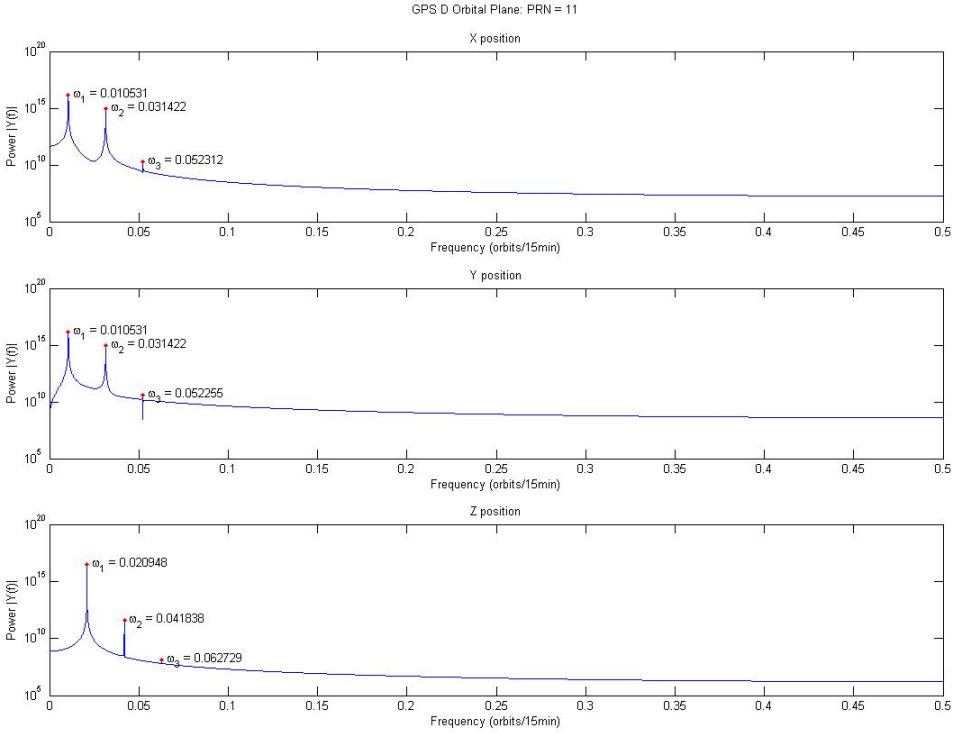

Figure 4.4: Frequencies of positions for PRN 11 located in the D orbital plane

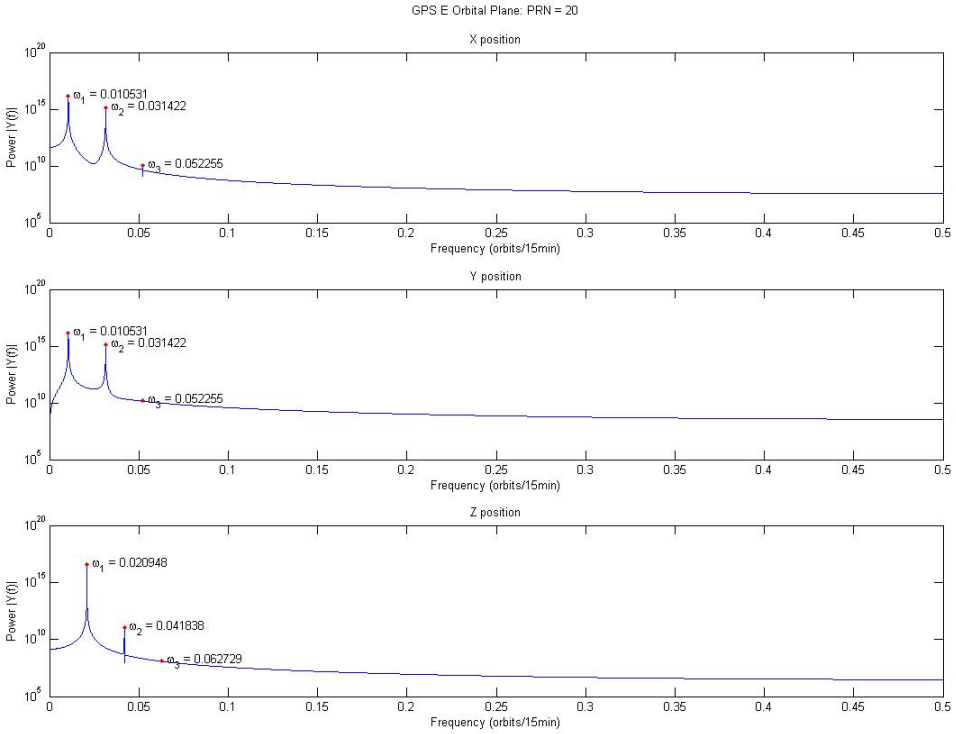

Figure 4.5: Frequencies of positions for PRN 20 located in the E orbital plane

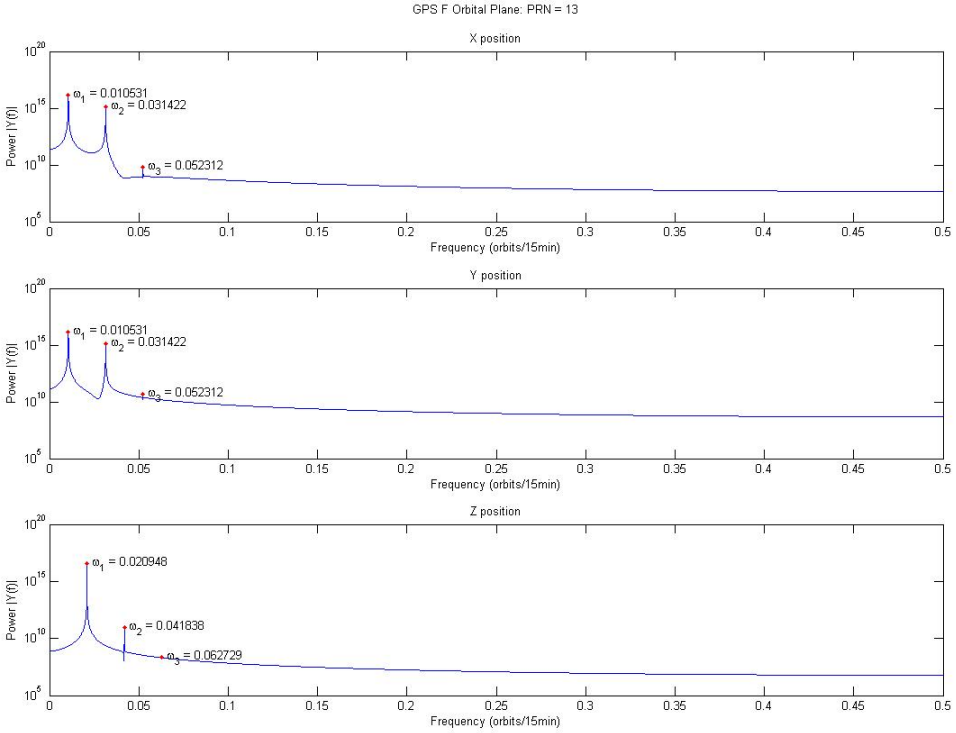

Figure 4.6: Frequencies of positions for PRN 13 located in the E orbital plane

The frequencies show some inconsistencies. In Figures 4.3 and 4.4, the third frequency in each of the x and y axes are not identical. Table 4.1 has all of the frequencies for each axis as shown in Figure 4.1 and includes the approximate identities relating each of the frequencies. The frequencies are almost multiples of each other but have some error. By direct calculation $2\omega_1 = 0.041896$ rather than 0.041838 as determined with the Fourier transformation. Similarly, by direct calculation $\omega_1 + \omega_2 = 0.031479$ rather than 0.031422 as determined with the Fourier transformation. ω_3 does not appear in any of the frequencies. Because the orbit is near circular, the apsidial regression rate is almost zero. In each of the graphs, the first frequency in the z coordinate bisects the first two frequencies in the x and y coordinates. The resonance between the orbital period and the rotation rate of the Earth results in $\omega_2 = 2\omega_1$. GPS satellites orbit the earth once every 12 hours and the Earth completes one revolution every 24 hours. This coupling essentially causes us to loose a frequency since ω_1 and ω_2 are multiples of each other rather than discrete unique frequencies.

The difference in results for the frequencies may be explained by the presence of other small magnitude frequencies that did not directly show up in the analysis. Though small, ω_3 may be buried in the results. Because this analysis was completed on actual satellite data, it is possible that the sun or moon may be affecting the orbits slightly. These interactions could be represented with their own small frequencies that are not readily apparent.

The oldest satellite currently in operation is PRN 25, located in the A orbital plane. It was launched in 1992. Analysis of this satellite produced interesting results. Figure 4.7 shows that the satellite is semi-stable.

Although PRN 25 shows distinct frequencies, it has noise between the frequencies. The frequencies are also shifted compared to those of all the other GPS satellites analyzed. PRN 25 corresponds to SVN 25 and it is the only satellite in the constellation with only three reaction wheels. To correct for this, regular momentum dumps are completed. These momentum dumps are very short duration small pulses (with order of magnitude comparable to a "mouse fart"). [*Bordner*, 2008] These brief changes in velocity are enough to influence the analysis. This reinforces the conditions for the KAM theory that

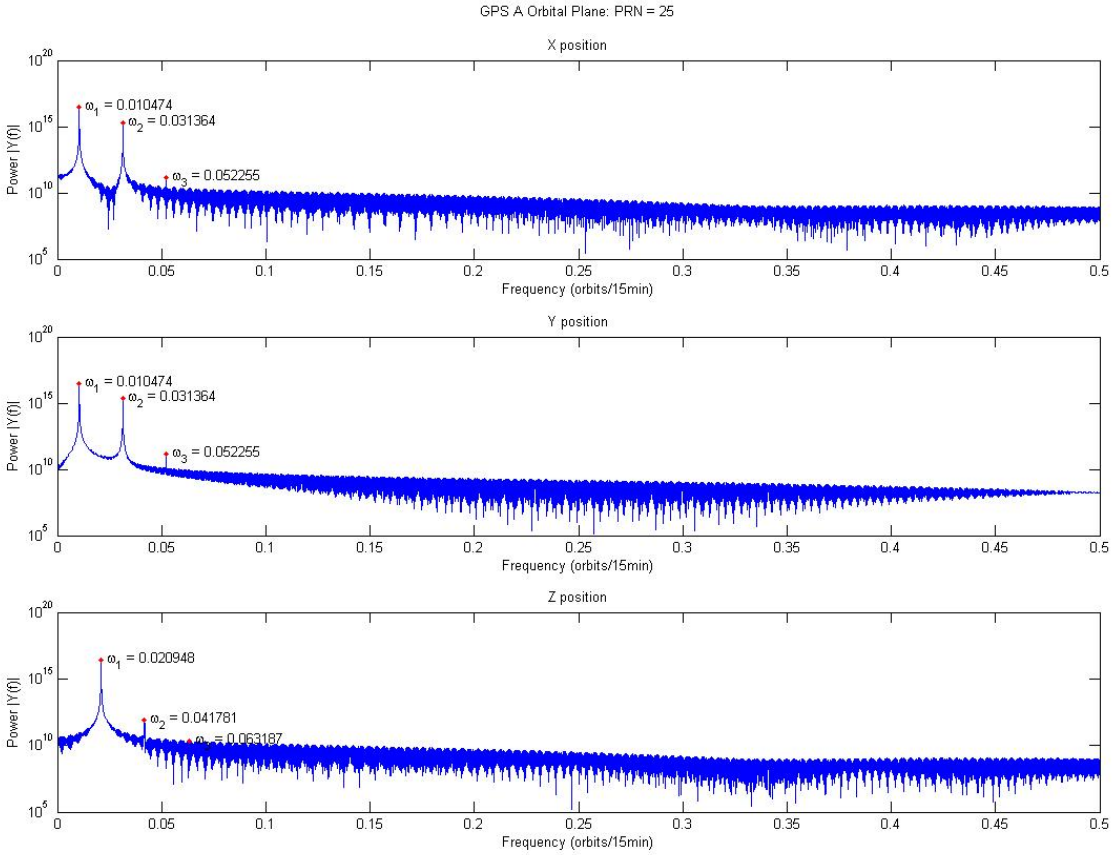

Figure 4.7: Frequencies of positions for PRN 25 located in the A orbital plane

all perturbations must be small and smooth. PRN 25 experiences small perturbations, but the burns by nature are not smooth changes. This satellite could still be modeled with a Fourier series representing the torus, but it is likely there would be greater error in the location predictions.

4.3 Integrated Orbital Frequencies

A numerical integration of the Hamiltian given in Section 2.3 using EGM 96 to order and degree n,m ¡ 20 was completed for a GPS satellite. The following values were used to begin the integration: x = 4.1642 DU, y = 0 DU, z = 0 DU in the ECEF frame and $\dot{x} = 0$ DU/TU, $\dot{y} = 0.2811$ DU/TU, $\dot{z} = 0.4014$ DU/TU in the ECI frame. These are based on an ideal satellite with e=0, i=55°, and a=26,560 km. The integration begins at the moment in time when the ECEF and ECI reference frames are aligned and the satellite is at the ascending node with RAAN = 0°. Figure 4.8 shows the error in the Hamiltonian over the course of the integration. The satellite orbit was integrated for 19,560 TU, which is approximately six months.

Figure 4.9 shows the frequencies identified for the numerically integrated orbit. The same patterns and approximate frequencies shown by the precise orbits also appear with the numerically integrated orbit. The frequencies for the precise orbits are given in orbits/15min, while the numerically integrated results are in canonical units of rad/TU. A simple conversion between these units shows the frequencies are the same and also correspond to the initial estimates in Section 4.1. Figure 4.10 shows the detail of the higher order frequencies for the numerically integrated orbit.

4.4 Laskar Frequency Fit

Using the Laskar frequency fitting algorithm described in Section 2.6, the orbital frequencies of the numerically integrated orbit are found to double precision. Table 4.2 details all the frequencies for each axis and the approximate identities they represent.

The frequencies identified with the Laskar frequency algorithm show the same patterns and interesting results found with the frequency fit of the precise orbit data. The x and y axis frequencies are close but do not match beyond two to six decimal places.

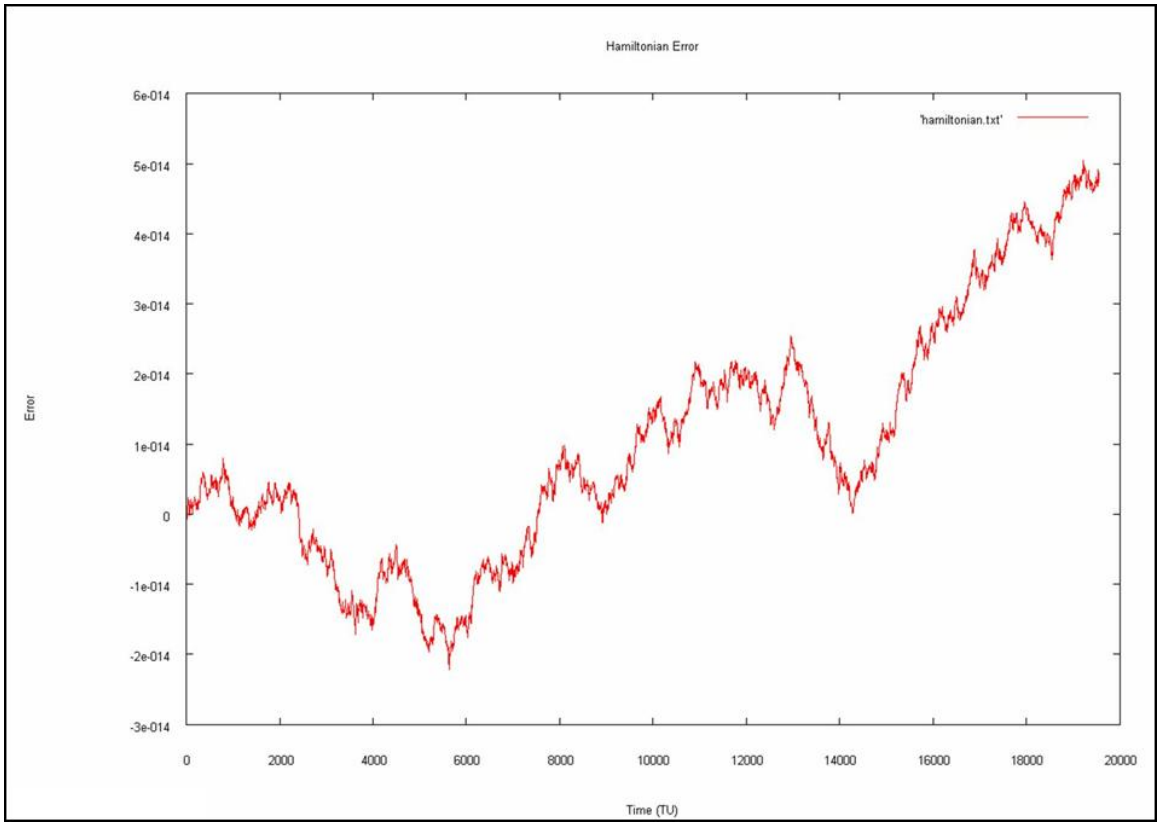

Figure 4.8: Hamiltonian Error of Numerically Integrated Orbit

Table 4.2: Numerically Integrated Orbital Frequencies using Laskar Frequency Fitting

Coordinate	Frequency $\frac{\text{rad}}{\text{TU}}$	Identity
x	5.885905920973412e-2	ω_2
y	5.885907961866142e-2	ω_2
z	1.17698982359873e-1	ω_1
x	1.75444154860682e-1	$\omega_1 + \omega_2$
y	1.75115819066544e-1	$\omega_1 + \omega_2$
z	2.35399935912444e-1	$2\omega_1$
x	2.94232047488148e-1	$\omega_2 + 2\omega_1$
y	2.91853937261849e-1	$\omega_2 + 2\omega_1$
z	3.53097100563892e-1	$3\omega_1$

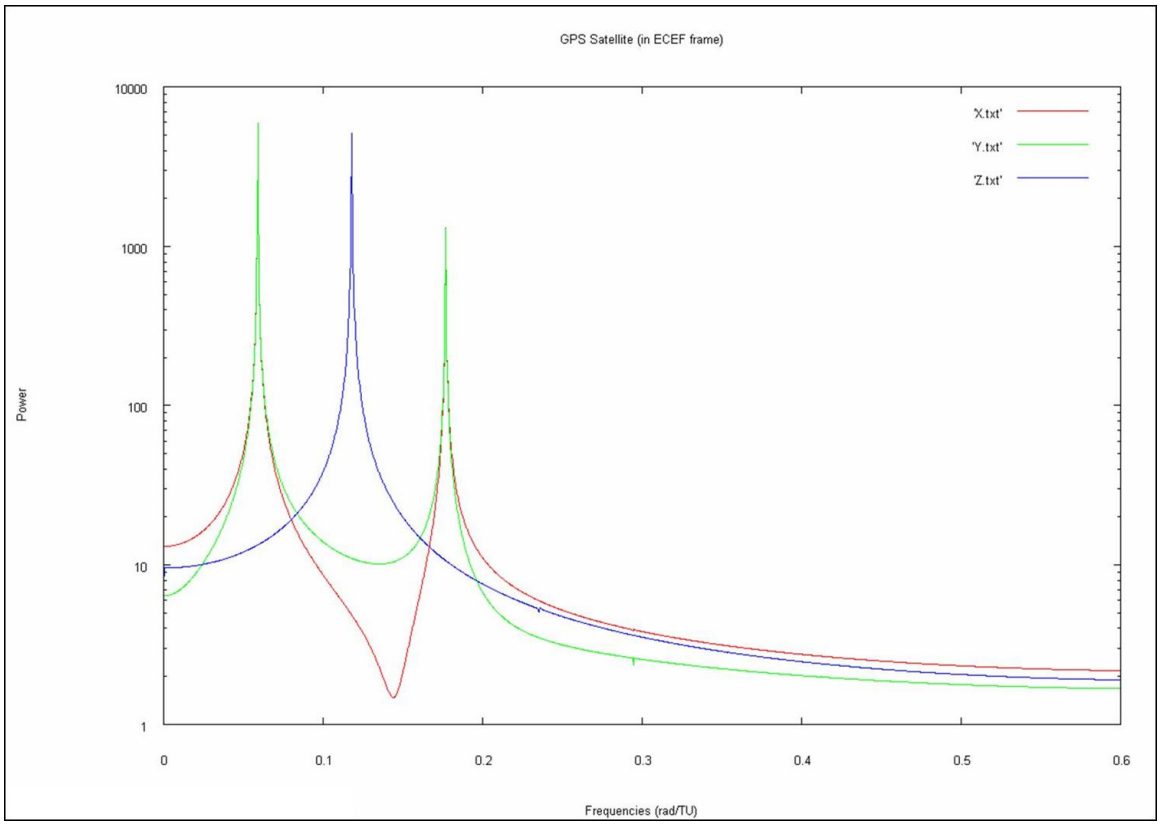

Figure 4.9: Frequencies of Integrated Orbit, 0-0.6 rad/TU

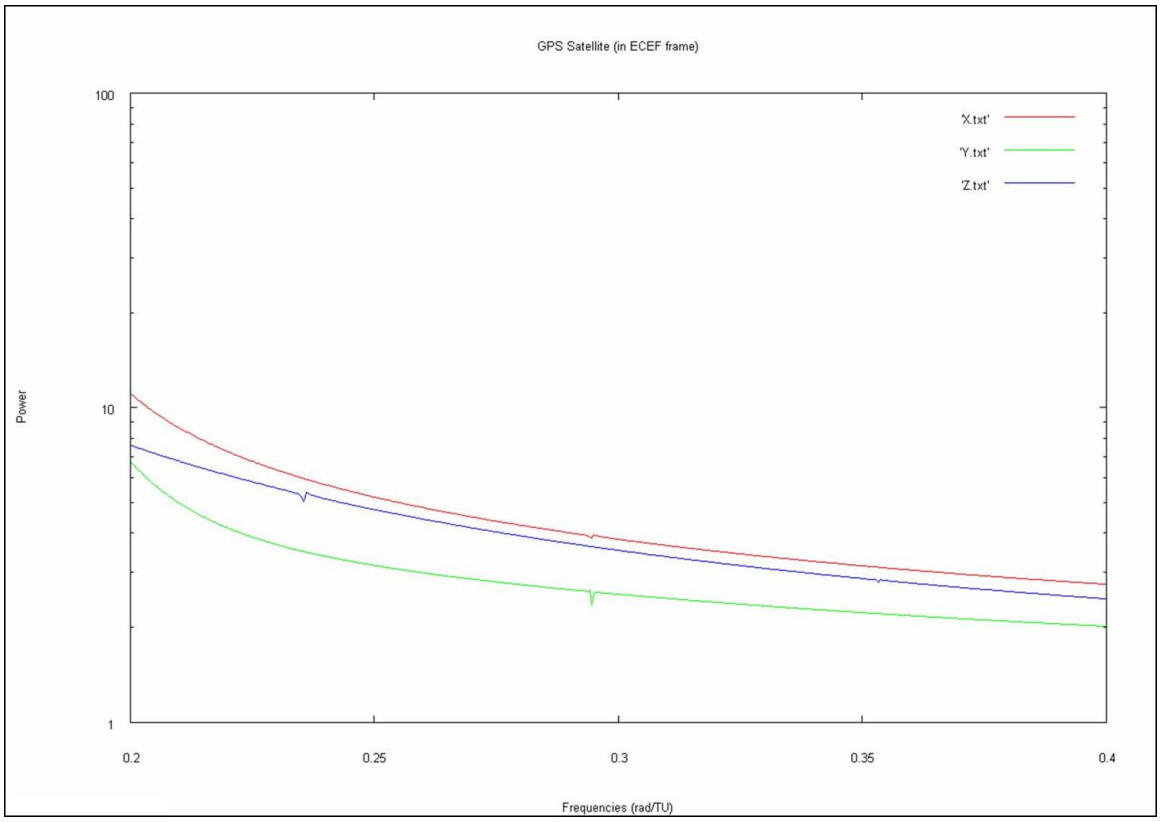

Figure 4.10: Frequencies of Integrated Orbit, 0.2-0.4 rad/TU

Furthermore, the identities are not exactly represented in the frequencies beyond three to five decimal places.

The difference in results for the frequencies may be explained by the presence of other small magnitude frequencies that did not directly show up in the analysis. As discussed in Section 4.2, ω_3, though small, may be buried in the results. Other errors may be a result of errors in the numerical integration or in the fitting process. Although the integration was run for six months, it does not represent motion throughout the entire torus. This may require that the integration to be run for a longer time period which would allow the Laskar algorithm to sample the entire torus in the frequency fitting process.

V. Conclusions

Chapter IV shows promising but inconclusive results. Further analysis is required to understand the inconsistencies in the data and ultimately to prove that KAM theory can accurately model actual satellite motion. To verify accuracy, the KAM tori should be used to predict future satellite positions and these positions should be compared to the actual positions. More advanced studies may look at the actual application of KAM theory to aid in challenging problems such as formation flying of satellites or rapidly reacquiring "lost" objects.

5.1 Recommendations for Further Study

First and foremost, the frequencies should be evaluated to understand the inconsistencies.

Once the frequencies are understood and accurately determined, the coefficients of the model should be fit using a linear least squares fitting. This complete equation can be used to predict future satellite locations. At a future time, these predicted locations can be compared to actual satellite positions to determine the error in the KAM tori model of the satellite orbit.

Two studies should be completed to understand the trade offs between numerical integration and KAM tori for predicting satellite orbits. First, since the Laskar frequency fitting algorithm is computationally intensive and time consuming to implement, there is little benefit to finding a KAM torus for an orbit if only a short period of time is required. For example, with the space shuttle it would be more beneficial to do a numerical integration. The Laskar frequency fitting for a KAM torus is a one time calculation. Once it is complete the computational and time requirements are minimal. Using a KAM torus to represent debris orbits over long periods of time would be beneficial. A study evaluating the cumulative computational and time requirements for a numerical integration versus a KAM tori fitting with prediction would give guidelines as to when each method should be used. A second study evaluating the effects of air drag on the KAM location predictions would give guidelines as to the minimum altitude for KAM tori to be applied. The Hamiltonian only includes the gravitational perturbation to the

satellite orbit. As a satellite's altitude decreases, air drag perturbations increase. KAM tori have been applied to low altitude satellites successfully. However, it is possible that for satellites in very low altitude orbits the perturbations from drag would be too great to apply KAM theory.

5.2 Application of KAM to Earth Orbiting Satellites

There are several situations where the application of KAM tori could be beneficial to the operation of Earth orbiting satellites. Specifying a KAM torus for a given orbit, a satellite's position is known at any point in the future, up to a limit which will need to be determined. This valid time limit will likely be on the order of months, since the torus is fit based on months of data. An orbit model that can directly calculate a satellite orbit at any point in time is extremely valuable. Once the KAM torus is identified there will be lower computational requirements for determining the position of a satellite and especially for determining multiple satellite locations simultaneously. Another benefit of this method would be that almanacs and broadcast ephemeris such as those used by GPS would be valid for longer periods of time.

Two satellites that are on the same or related tori remain in the same relative position to each other. This method could be used to set up formation flying of satellites instead of using the Clohessy-Wilshire equations.

In the case of an orbit that experiences a sudden change of trajectory, KAM theory could be applied up to the impulse. Keplarian calculations could then be used to calculate the orbit following the impulse. Subsequently, the orbital elements from the Keplarian solution could be used to estimate the frequencies of the new orbit. These may be able to be used to predict the approximate satellite location, thus allowing tracking systems to reacquire the "lost" object.

Appendix A. Constants and GPS Data

A.1 GPS Parameter Summary and Constants

[*Misra and Enge*, 2001]

Table A.1: GPS Constellation Parameters

Parameter	Nominal Value	Tolerance
a	26,560 km	+/- 50 km
e	less than 0.02	n/a
i	55 deg	+/- 3 deg
Period	11 hr 58 min	
Operational Satellites	24	+8
Planes	6	n/a
RAAN spacing	60 deg at equator	n/a
Satellites per plane	4	+1
Inter-satellite spacing	2@30-32.1deg	

A.2 Earth Constants

[*Bate et al.*, 1971]

Table A.2: Geocentric Constants

Geocentric Parameter	Canonical Units	Metric Units
Mean Equatorial Radius, $r_\oplus$	1 DU	6378.145 km
Time Unit	1 TU	806.8118744 sec
Speed Unit	1 $\frac{DU}{TU}$	7.90536828 $\frac{km}{sec}$
Gravitational Parameter, $\mu_\oplus$	1 $\frac{DU^3}{TU^2}$	3.986012e5 $\frac{km^3}{sec^2}$
Angular Rotation, $\omega_\oplus$	.0588336565 $\frac{rad}{TU}$	7.292115856e-5 $\frac{rad}{sec}$

Appendix B. 2007 GPS Constellation Frequencies

The following graphs are of the frequency and power of the orbits in each of the ECEF coordinates. The graphs represent all of the satellites that were in operation for all of 2007. The frequency analysis was done for January through June.

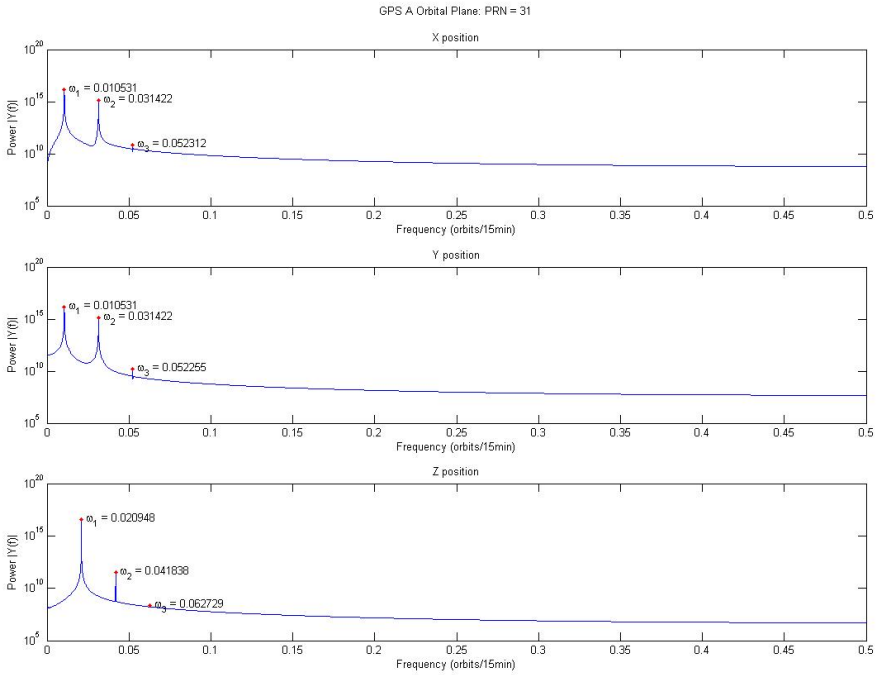

Figure B.1: Frequencies of positions for PRN 31 located in the A orbital plane

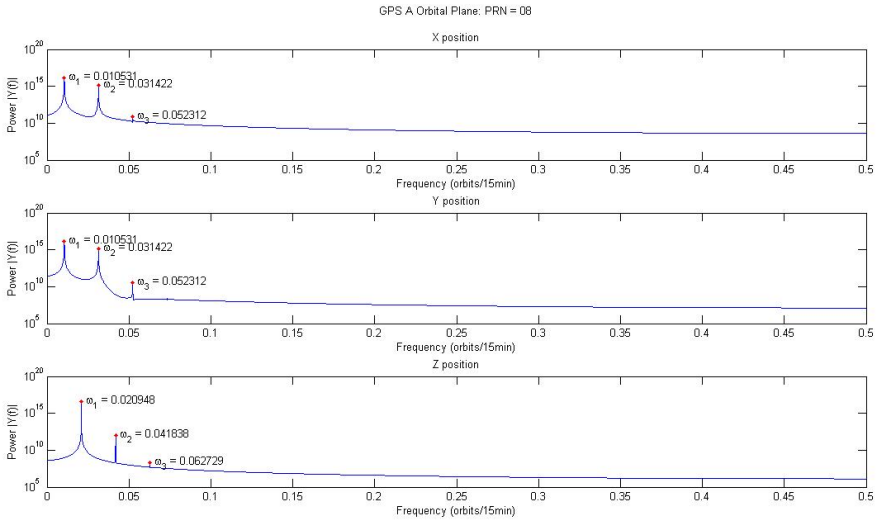

Figure B.2: Frequencies of positions for PRN 08 located in the A orbital plane

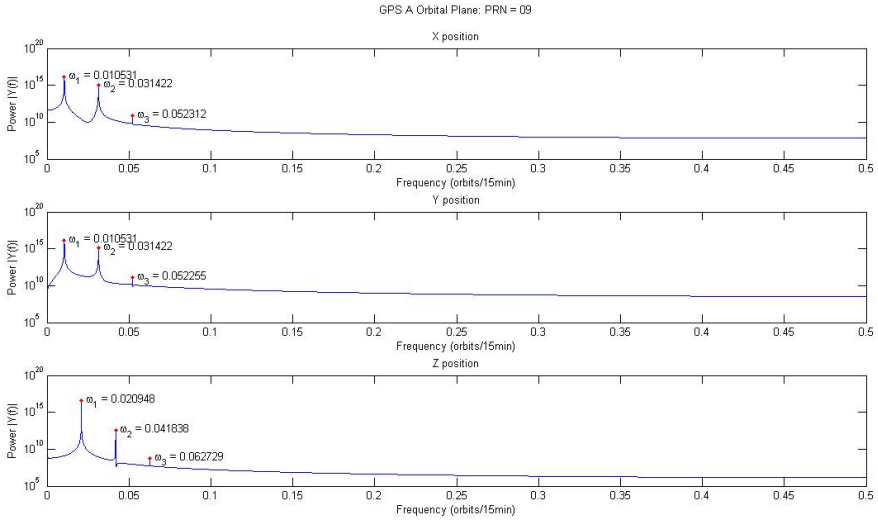

Figure B.3: Frequencies of positions for PRN 09 located in the A orbital plane

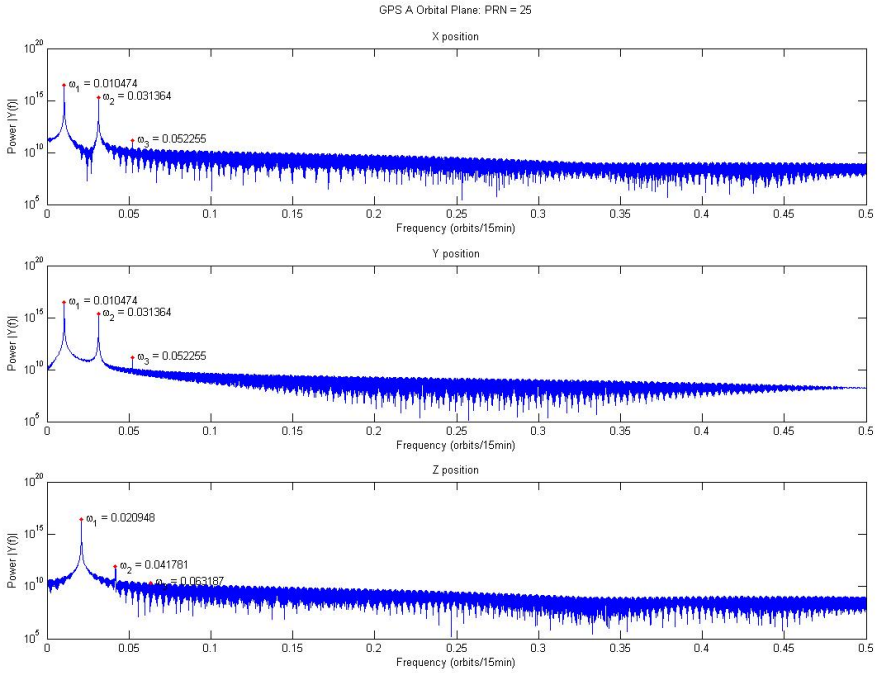

Figure B.4: Frequencies of positions for PRN 25 located in the A orbital plane

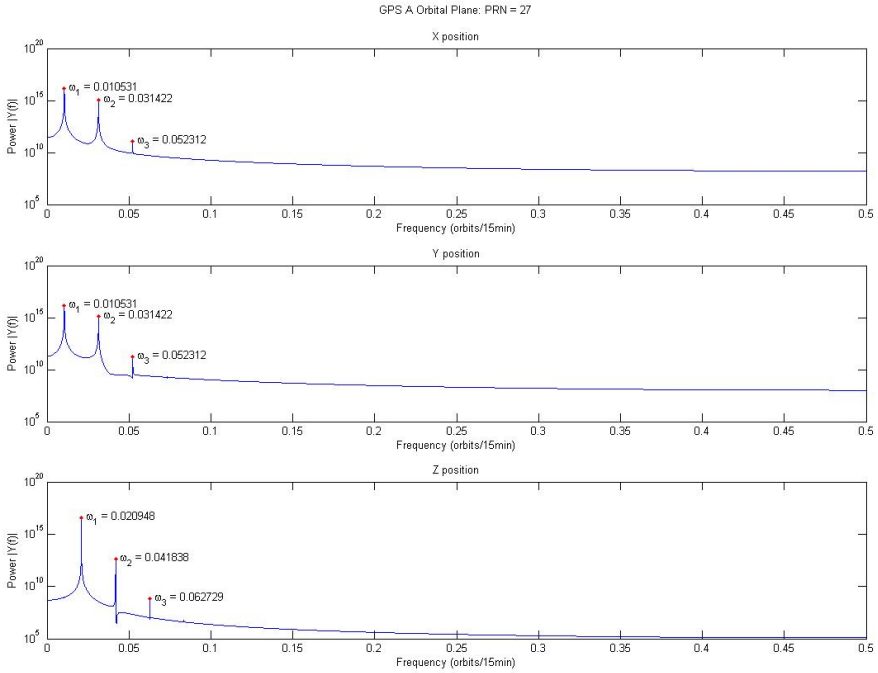

Figure B.5: Frequencies of positions for PRN 27 located in the A orbital plane

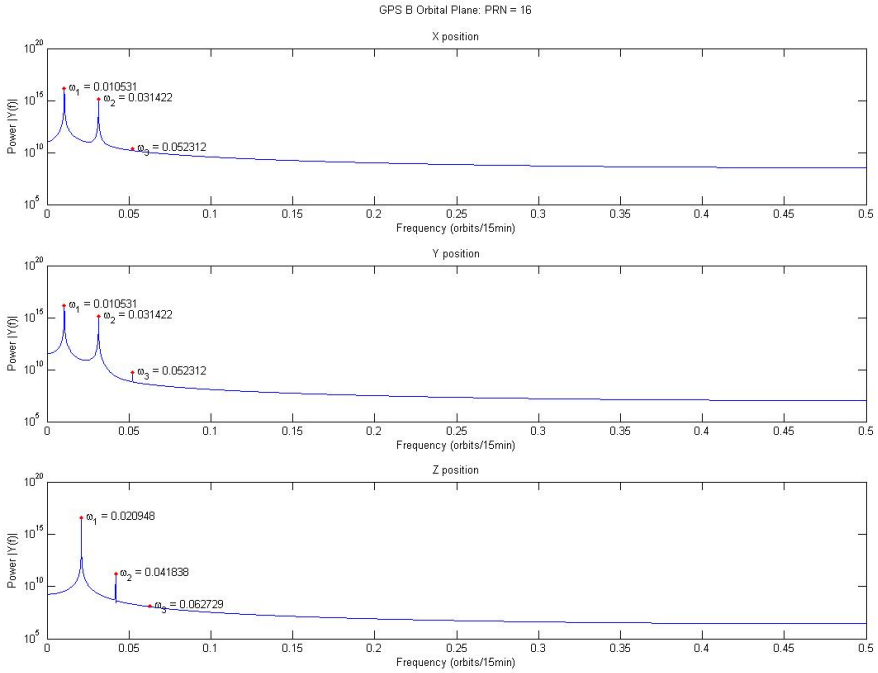

Figure B.6: Frequencies of positions for PRN 16 located in the B orbital plane

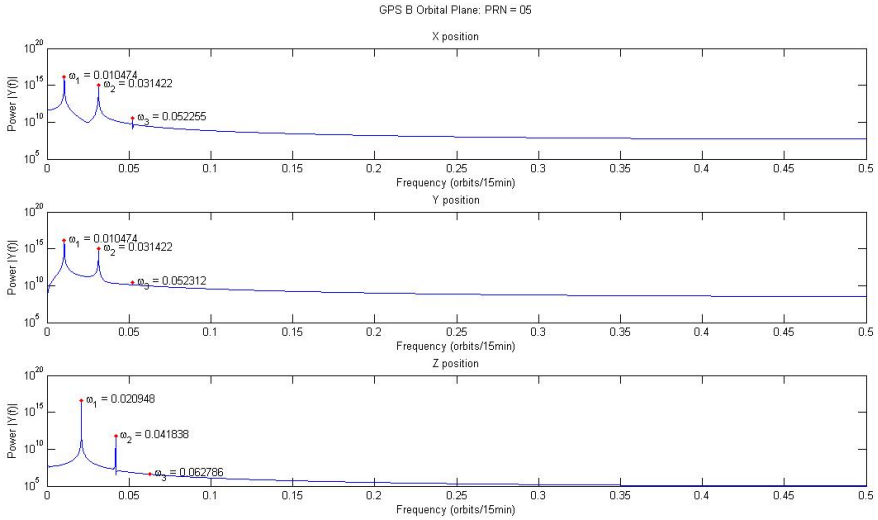

Figure B.7: Frequencies of positions for PRN 05 located in the B orbital plane

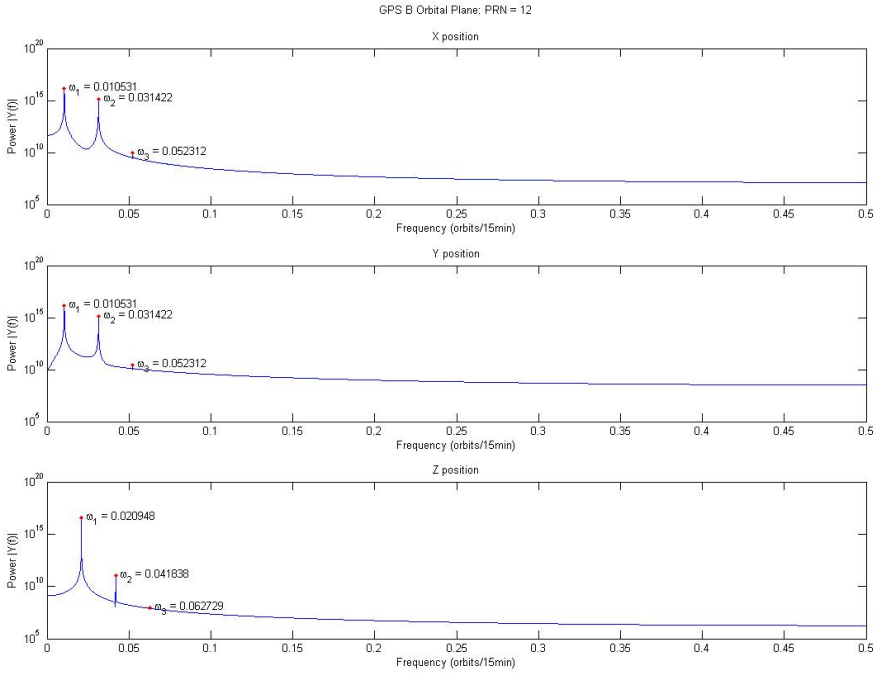

Figure B.8: Frequencies of positions for PRN 12 located in the B orbital plane

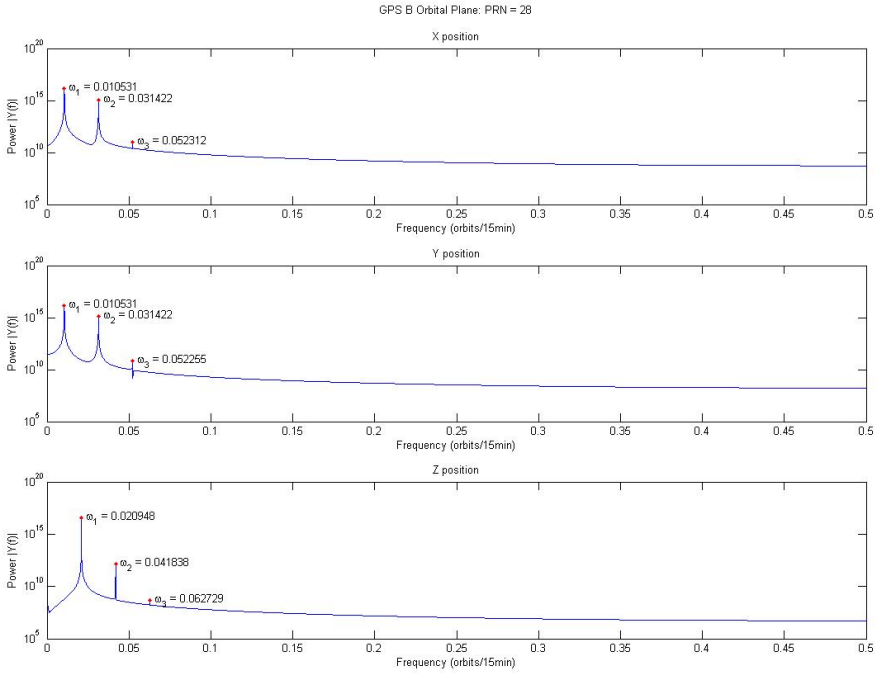

Figure B.9: Frequencies of positions for PRN 28 located in the B orbital plane

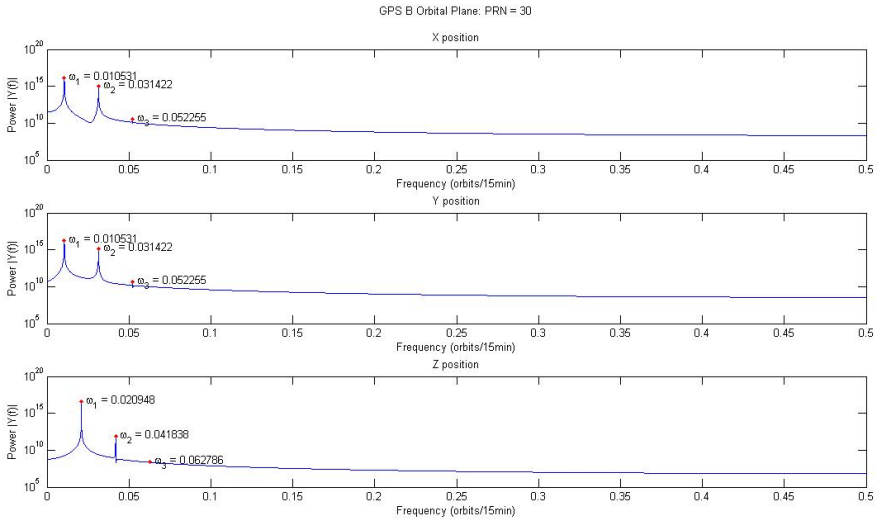

Figure B.10: Frequencies of positions for PRN 30 located in the B orbital plane

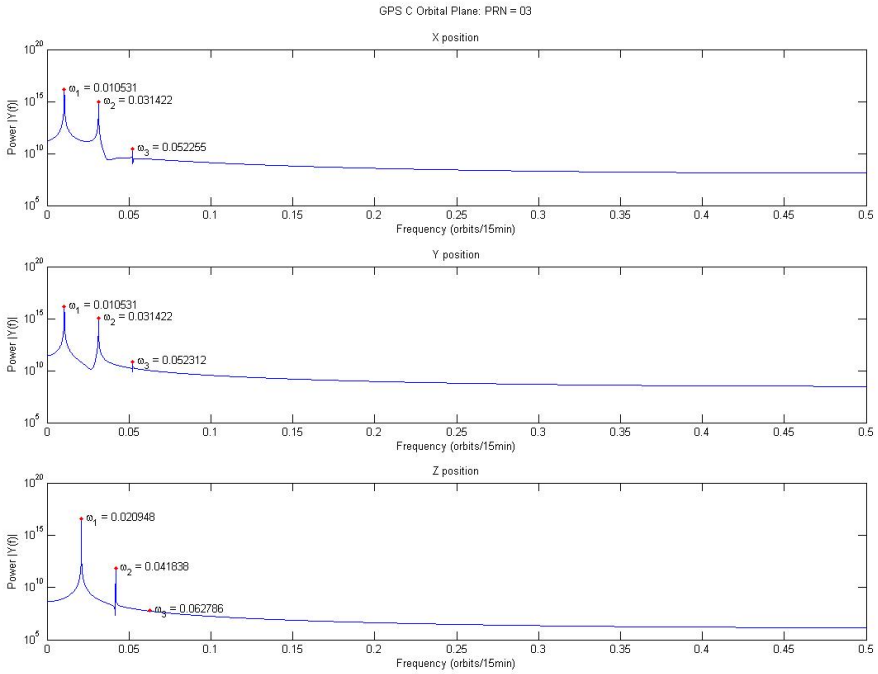

Figure B.11: Frequencies of positions for PRN 03 located in the C orbital plane

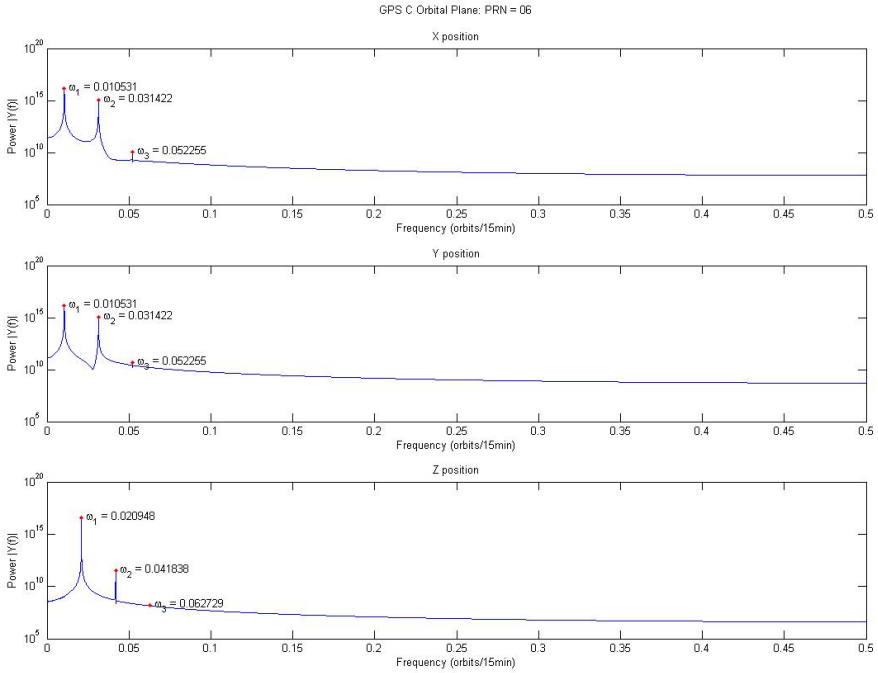

Figure B.12: Frequencies of positions for PRN 06 located in the C orbital plane

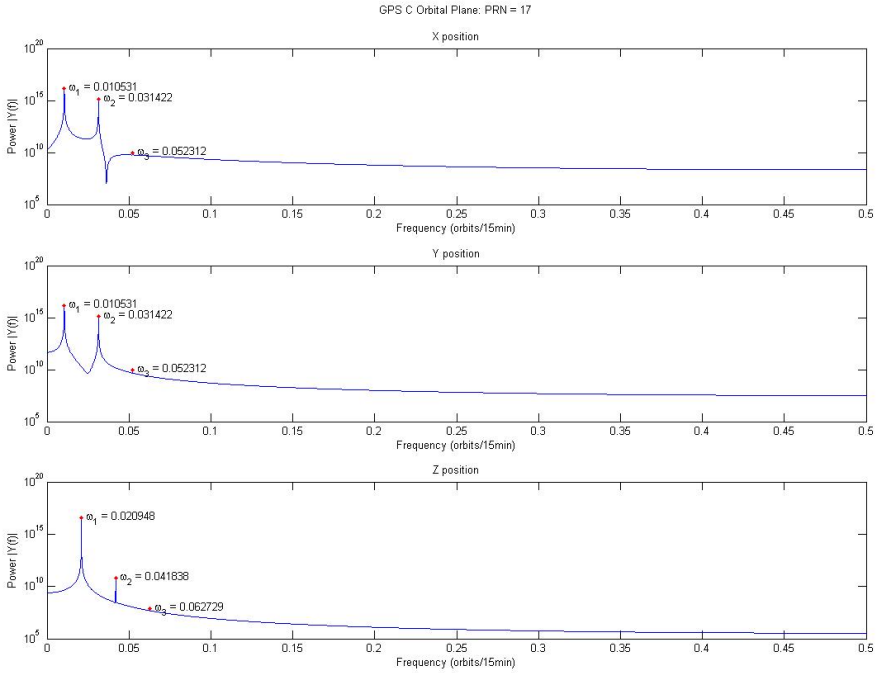

Figure B.13: Frequencies of positions for PRN 17 located in the C orbital plane

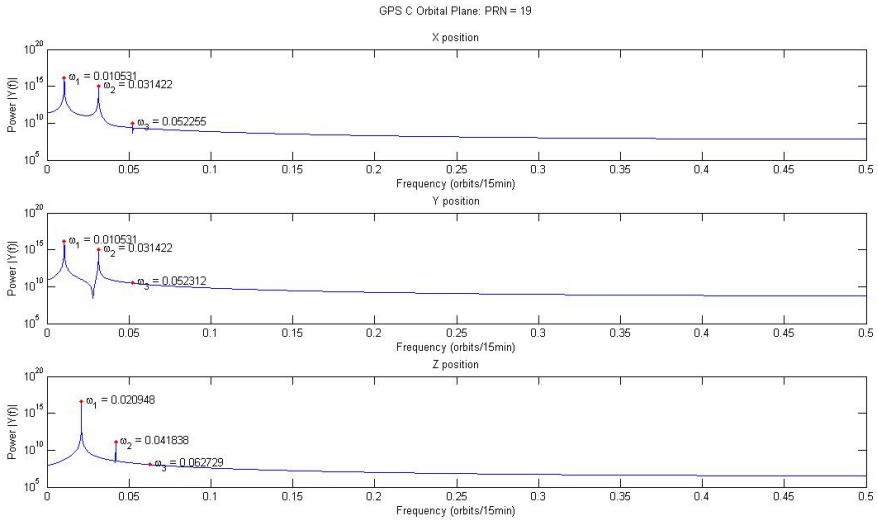

Figure B.14: Frequencies of positions for PRN 19 located in the C orbital plane

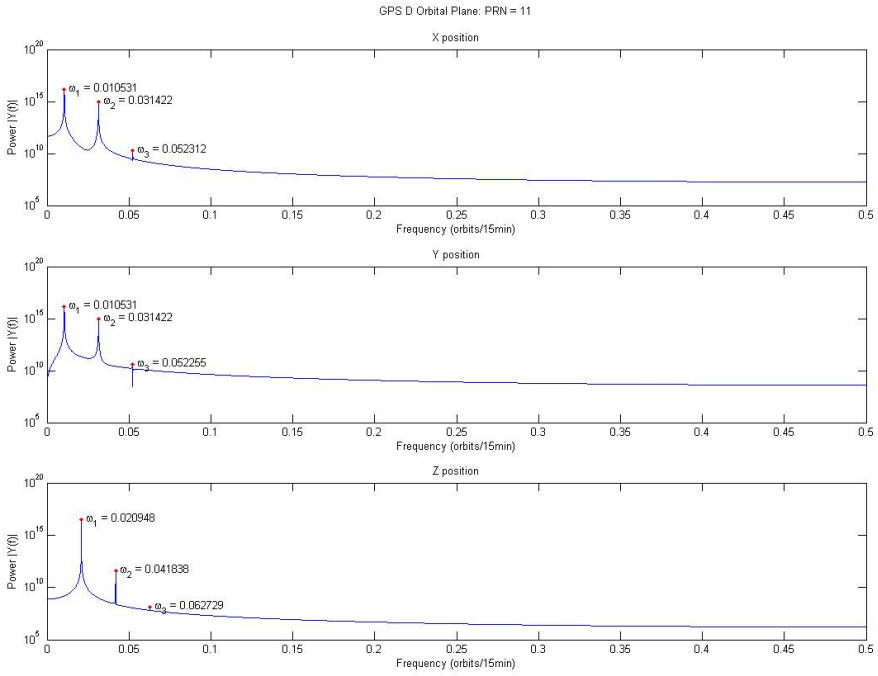

Figure B.15: Frequencies of positions for PRN 11 located in the D orbital plane

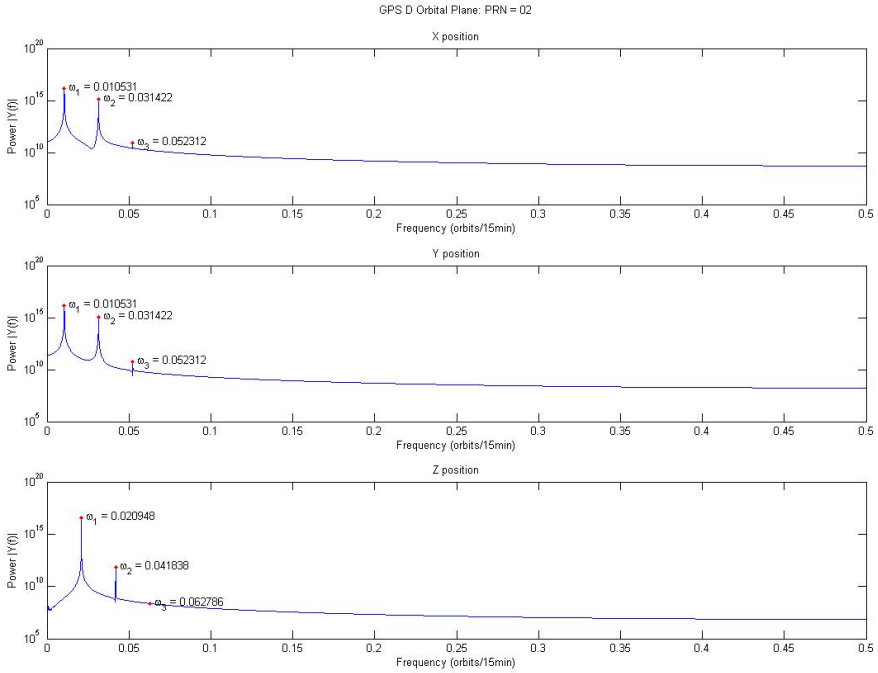

Figure B.16: Frequencies of positions for PRN 02 located in the D orbital plane

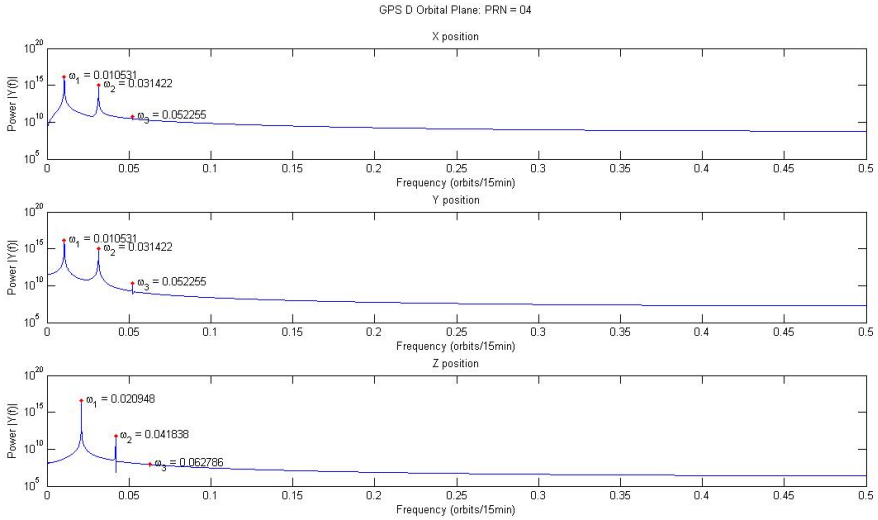

Figure B.17: Frequencies of positions for PRN 04 located in the D orbital plane

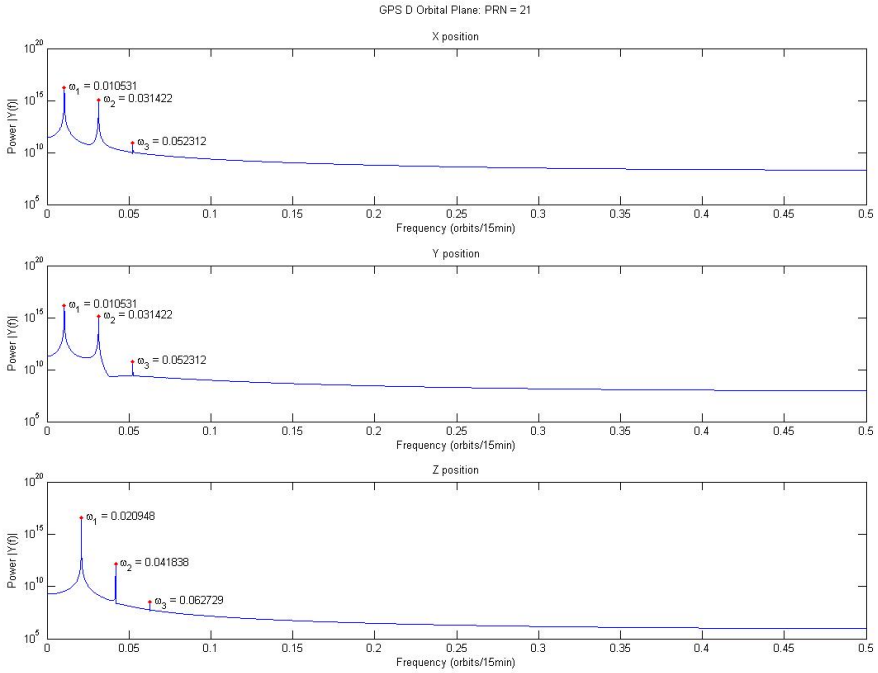

Figure B.18: Frequencies of positions for PRN 21 located in the D orbital plane

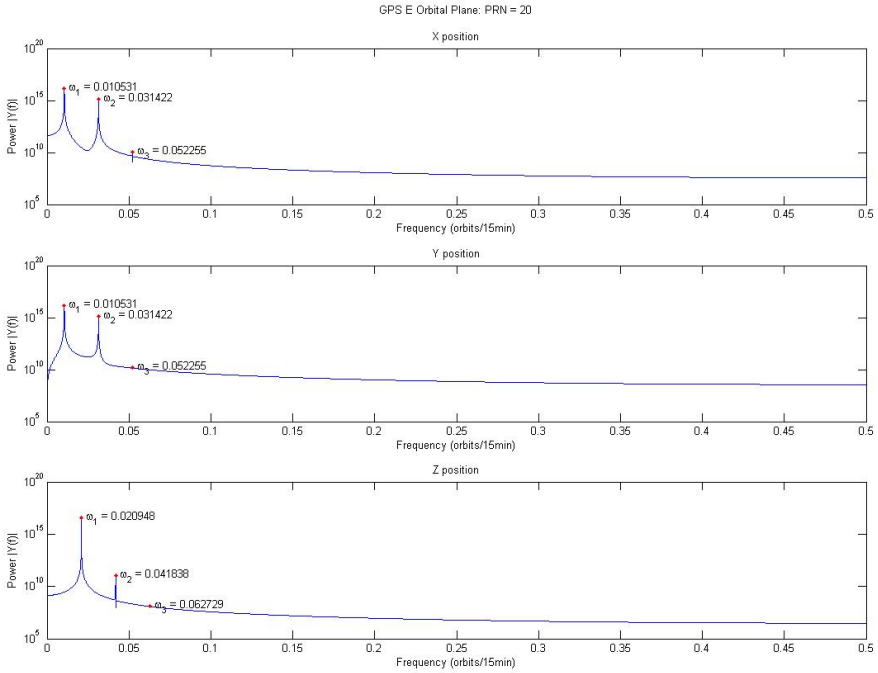

Figure B.19: Frequencies of positions for PRN 20 located in the E orbital plane

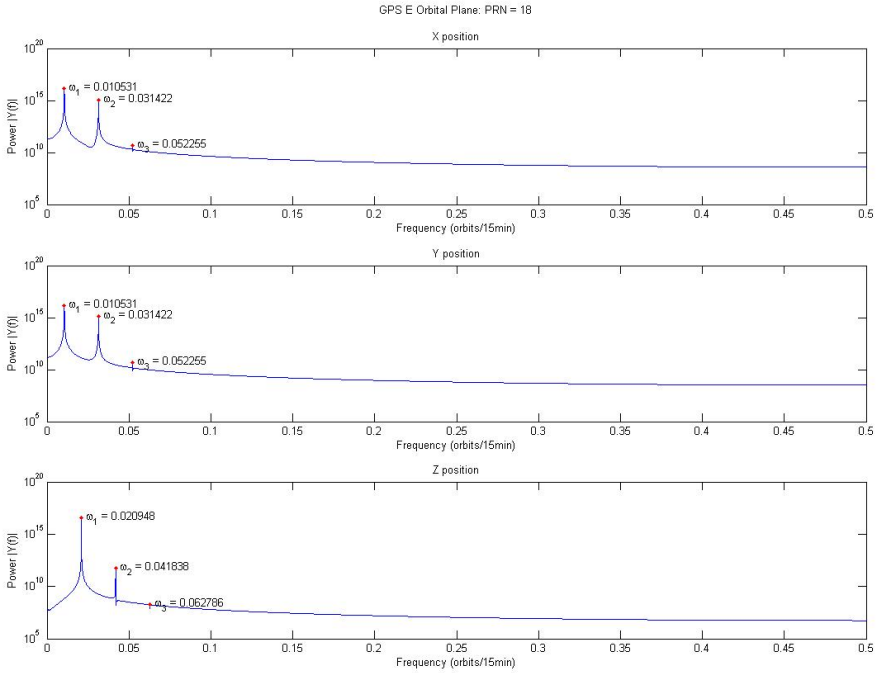

Figure B.20: Frequencies of positions for PRN 18 located in the E orbital plane

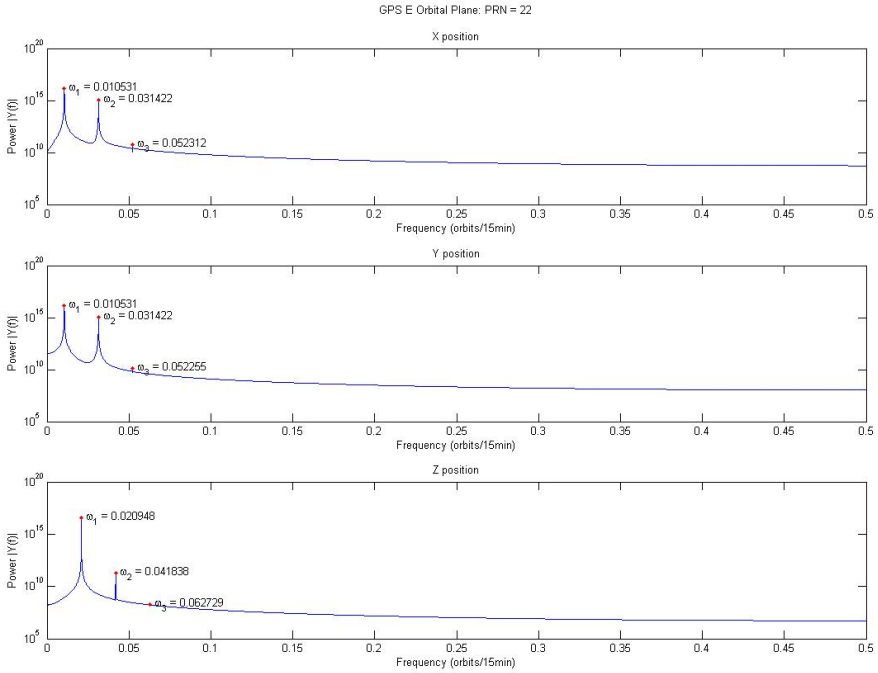

Figure B.21: Frequencies of positions for PRN 22 located in the E orbital plane

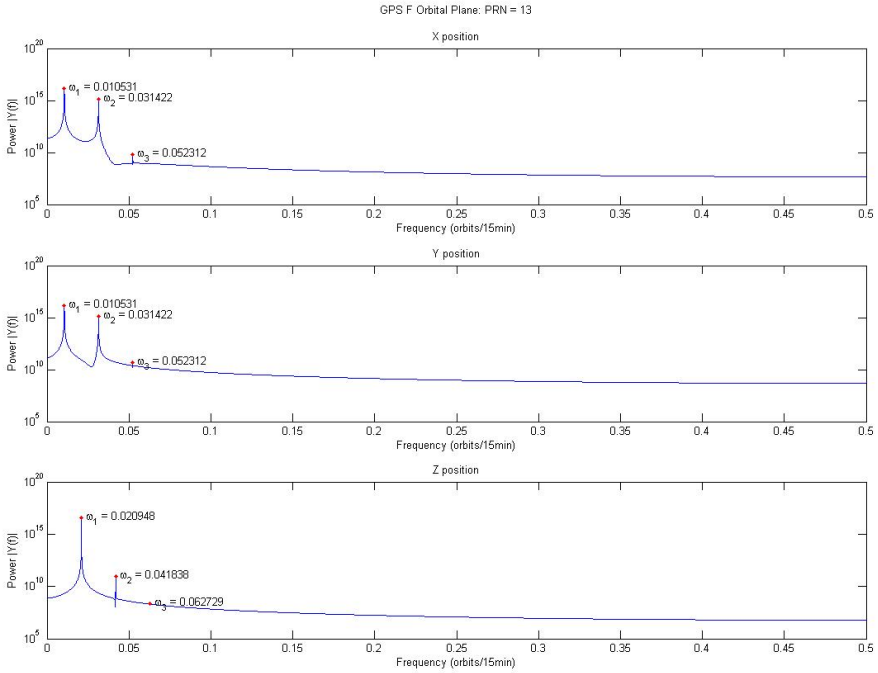

Figure B.22: Frequencies of positions for PRN 13 located in the F orbital plane

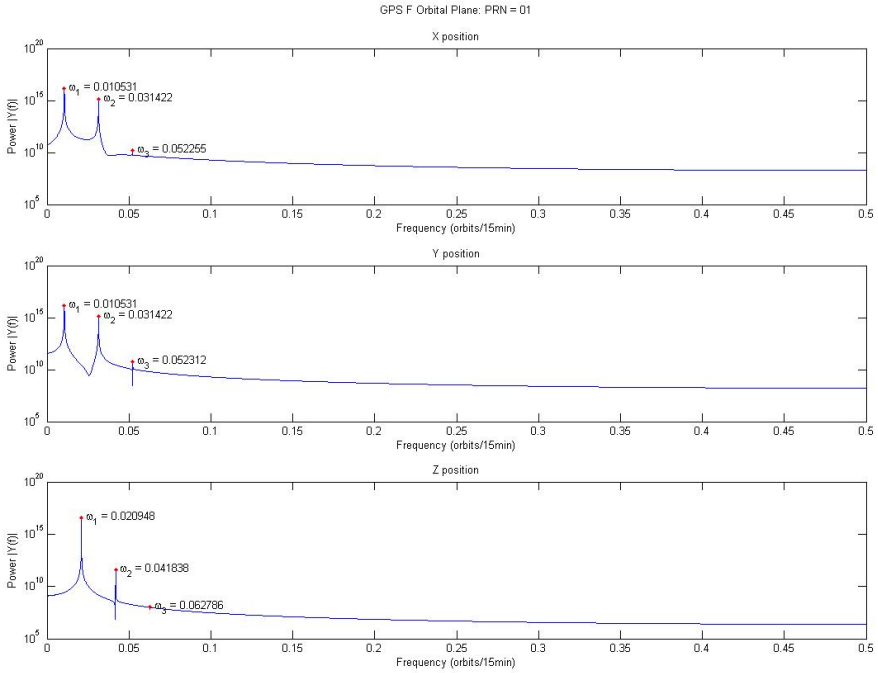

Figure B.23: Frequencies of positions for PRN 01 located in the F orbital plane

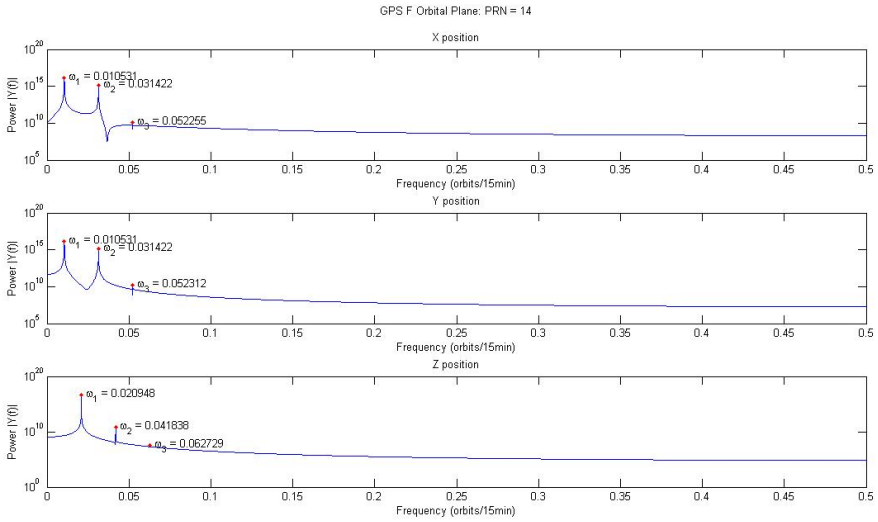

Figure B.24: Frequencies of positions for PRN 14 located in the F orbital plane

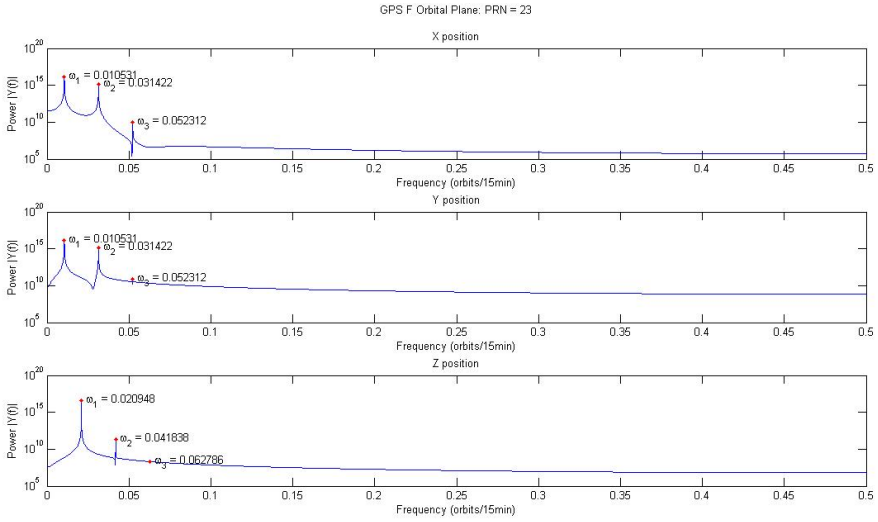

Figure B.25: Frequencies of positions for PRN 23 located in the F orbital plane

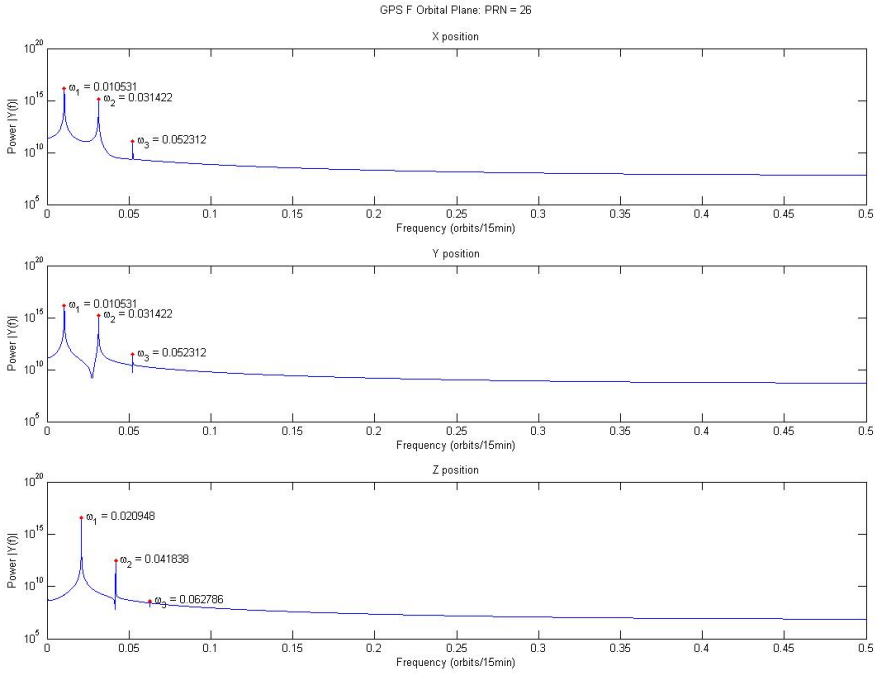

Figure B.26: Frequencies of positions for PRN 26 located in the F orbital plane

The following code files were written in Matlab version 2007b for analysis of the precise satellite orbit data. Dr William Wiesel has developed Fortran 90 code to do the numerically integrated orbit and the frequency identification of this orbit.

C.1 Main Data Analysis File

Listing C.1: Main Data Analysis File

```
%Capt Rachel Derbis
%Main Thesis Script
%Version 7.2

%This script will take precise matlab orbits and calculate the ...
    orbital
%frequencies

%This work is based on the KAM theory.
%All position and velocity values are in the earth centered earth ...
    fixed
%(rotating) reference frame unless otherwise notes as the earth ...
    centered
%inertial frame

%clear matlab to start new session
clear
clc
format long e

%Constants for Earth(from Fundamentals of Astrodynamics p429)
%Metric Units
mu = 3.986012e5; %Gravitational Parameter (km^3/sec^2)
Re = 6378.145; %Mean Equatorial Radius (km)
omega = 7.292115856e-5; %Angular Rotaton (rad/sec)
tu = 806.8118744; %Time Unit (sec)
su = 7.90536828; %Speed Unit (km/sec)
%Canonical Units
mu_c = 1; %Gravitational Parameter (DU^3/TU^2)
Re_c = 1; %Mean Equatorial Radius (DU)
omega_c = .0588336565; %Angular Rotaton (rad/TU)
tu_c = 1; %Time Unit (TU)
su_c = 1; %Speed Unit (DU/TU)

%Select PRN of interest (this is the satellite considered)
%PRNs listed are for satellites fully operational for ALL of 2007
%A orbital plane includes: 25,27,09,08,31
%B orbital plane includes: 05,30,28,16,12
%C orbital plane includes: 06,03,19,17
%D orbital plane includes: 04,11,21,02
%E orbital plane includes: 20,18,22
%F orbital plane includes: 26,01,13,14,23

```

```
%set the number for the satellite grouping to analyze
%(this should be the only change between runs unless you have ...
   already
%sorted the data and only are doing analysis, then comment the code ...
   noted)
setnum = 1; %value between 1 and 5;
%first set of satellites produced interesting results, try other ...
   sets
%notice if there is not a satellite to be analyzed in a plane for a ...
   set
%then PRN = 00, this will produce messages stating this and dummy ...
   outputs
if setnum ==1;
    PRNa = 25;
    PRNb = 16;
    PRNc = 03;
    PRNd = 02;
    PRNe = 20;
    PRNf = 13;
elseif setnum ==2;
    PRNa = 31;
    PRNb = 12;
    PRNc = 17;
    PRNd = 11;
    PRNe = 18;
    PRNf = 01;
elseif setnum ==3;
    PRNa = 27;
    PRNb = 28;
    PRNc = 06;
    PRNd = 21;
    PRNe = 22;
    PRNf = 26;
elseif setnum ==4;
    PRNa = 09;
    PRNb = 30;
    PRNc = 19;
    PRNd = 04;
    PRNe = 00;
    PRNf = 14;
elseif setnum ==5;
    PRNa = 08;
    PRNb = 05;
    PRNc = 00;
    PRNd = 00;
    PRNe = 00;
    PRNf = 23;
end

%initialize filenames for saving and recalling, based on setnum
filename2 = ['orbit', num2str(setnum)];
filename4 = ['velocities', num2str(setnum)];
filename5 = ['canonical', num2str(setnum)];

%**A********comment out code here if data has been presorted******
```

```
%Read in the final orbit data(sp3 file)
filename1 = 'nd6Mon.sp3';
disp 'Building time and position matrices from precise GPS file'
%A orbital plane read in data
[xp_a, yp_a, zp_a, time_a] = pos_final(PRNa, filename1);
disp 'checkpoint A1'
%B orbital plane read in data
[xp_b, yp_b, zp_b, time_b] = pos_final(PRNb, filename1);
disp 'checkpoint B1'
%C orbital plane read in data
[xp_c, yp_c, zp_c, time_c] = pos_final(PRNc, filename1);
disp 'checkpoint C1'
%D orbital plane read in data
[xp_d, yp_d, zp_d, time_d] = pos_final(PRNd, filename1);
disp 'checkpoint D1'
%E orbital plane read in data
[xp_e, yp_e, zp_e, time_e] = pos_final(PRNe, filename1);
disp 'checkpoint E1'
%F orbital plane read in data
[xp_f, yp_f, zp_f, time_f] = pos_final(PRNf, filename1);
disp 'checkpoint F1'

%save orbit for future use.
orbit_a = [xp_a;yp_a;zp_a;time_a];
orbit_b = [xp_b;yp_b;zp_b;time_b];
orbit_c = [xp_c;yp_c;zp_c;time_c];
orbit_d = [xp_d;yp_d;zp_d;time_d];
orbit_e = [xp_e;yp_e;zp_e;time_e];
orbit_f = [xp_f;yp_f;zp_f;time_f];
save (filename2,'orbit_a','orbit_b','orbit_c','orbit_d',...
    'orbit_e','orbit_f')

%**** to load existing presorted file begin here****
%comment out section of code starting with **A** above
%**B**if files were not sorted above begin comment out this section
%{
if setnum == 1;
    load orbit1.mat
elseif setnum == 2;
    load orbit2.mat
elseif setnum == 3;
    load orbit3.mat
elseif setnum == 4;
    load orbit4.mat
elseif setnum == 5;
    load orbit5.mat
end

%extract data from orbit file
xp_a = orbit_a(1,:);
yp_a = orbit_a(2,:);
zp_a = orbit_a(3,:);
time_a = orbit_a(4,:);
xp_b = orbit_b(1,:);
```

```
    yp_b = orbit_b(2,:);
    zp_b = orbit_b(3,:);
    time_b = orbit_b(4,:);
    xp_c = orbit_c(1,:);
150 yp_c = orbit_c(2,:);
    zp_c = orbit_c(3,:);
    time_c = orbit_c(4,:);
    xp_d = orbit_d(1,:);
    yp_d = orbit_d(2,:);
155 zp_d = orbit_d(3,:);
    time_d = orbit_d(4,:);
    xp_e = orbit_e(1,:);
    yp_e = orbit_e(2,:);
    zp_e = orbit_e(3,:);
160 time_e = orbit_e(4,:);
    xp_f = orbit_f(1,:);
    yp_f = orbit_f(2,:);
    zp_f = orbit_f(3,:);
    time_f = orbit_f(4,:);
165 %}
    %***comment out beginning at **B** if data is sorted in this run.
    %this is the end of the section that loads existing data

    %***End of Data input, beginning data analysis
170
    %do fast forier transform on each component A orbital plane
    disp 'Calculating FFT for each position matrix'
    Yax = fft(xp_a);
    Yay = fft(yp_a);
175 Yaz = fft(zp_a);
    disp 'checkpoint A2'
    %do fast forier transform on each component B orbital plane
    Ybx = fft(xp_b);
    Yby = fft(yp_b);
180 Ybz = fft(zp_b);
    disp 'checkpoint B2'
    %do fast forier transform on each component C orbital plane
    Ycx = fft(xp_c);
    Ycy = fft(yp_c);
185 Ycz = fft(zp_c);
    disp 'checkpoint C2'
    %do fast forier transform on each component D orbital plane
    Ydx = fft(xp_d);
    Ydy = fft(yp_d);
190 Ydz = fft(zp_d);
    disp 'checkpoint D2'
    %do fast forier transform on each component E orbital plane
    Yex = fft(xp_e);
    Yey = fft(yp_e);
195 Yez = fft(zp_e);
    disp 'checkpoint E2'
    %do fast forier transform on each component F orbital plane
    Yfx = fft(xp_f);
    Yfy = fft(yp_f);
200 Yfz = fft(zp_f);
```

```
    disp 'checkpoint F2'

    %Plot frequencies for each of the planes
    %A orbital plane plot and determine frequencies
205 Plane = 'A';
    PRN = num2str(PRNa,'%02d');
    [mFreqax, mFreqay, mFreqaz] = pfplot(Plane,PRN,Yax,Yay,Yaz);
    %B orbital plane plot and determine frequencies
    Plane = 'B';
210 PRN = num2str(PRNb,'%02d');
    [mFreqbx, mFreqby, mFreqbz] = pfplot(Plane,PRN,Ybx,Yby,Ybz);
    %C orbital plane plot and determine frequencies
    Plane = 'C';
    PRN = num2str(PRNc,'%02d');
215 [mFreqcx, mFreqcy, mFreqcz] = pfplot(Plane,PRN,Ycx,Ycy,Ycz);
    %D orbital plane plot and determine frequencies
    Plane = 'D';
    PRN = num2str(PRNd,'%02d');
    [mFreqdx, mFreqdy, mFreqdz] = pfplot(Plane,PRN,Ydx,Ydy,Ydz);
220 %E orbital plane plot and determine frequencies
    Plane = 'E';
    PRN = num2str(PRNe,'%02d');
    [mFreqex, mFreqey, mFreqez] = pfplot(Plane,PRN,Yex,Yey,Yez);
    %F orbital plane plot and determine frequencies
225 Plane = 'F';
    PRN = num2str(PRNf,'%02d');
    [mFreqfx, mFreqfy, mFreqfz] = pfplot(Plane,PRN,Yfx,Yfy,Yfz);

    %**C*** this section calculates the velocities from a brdc file
230
    %calculate the velocities
    filename3 = 'brdc6mon.07n';
    disp 'Building time and calculated velocity matrices from ephemeris'
    [xv_a, yv_a, zv_a, timev_a] = vel_brdc(PRNa, filename3);
235 disp 'checkpoint A3'
    [xv_b, yv_b, zv_b, timev_b] = vel_brdc(PRNb, filename3);
    disp 'checkpoint B3'
    [xv_c, yv_c, zv_c, timev_c] = vel_brdc(PRNc, filename3);
    disp 'checkpoint C3'
240 [xv_d, yv_d, zv_d, timev_d] = vel_brdc(PRNd, filename3);
    disp 'checkpoint D3'
    [xv_e, yv_e, zv_e, timev_e] = vel_brdc(PRNe, filename3);
    disp 'checkpoint E3'
    [xv_f, yv_f, zv_f, timev_f] = vel_brdc(PRNf, filename3);
245 disp 'checkpoint F3'

    %save velocities for future use.
    vel_a = [xv_a;yv_a;zv_a;timev_a];
    vel_b = [xv_b;yv_b;zv_b;timev_b];
250 vel_c = [xv_c;yv_c;zv_c;timev_c];
    vel_d = [xv_d;yv_d;zv_d;timev_d];
    vel_e = [xv_e;yv_e;zv_e;timev_e];
    vel_f = [xv_f;yv_f;zv_f;timev_f];
    save (filename4,'vel_a','vel_b','vel_c','vel_d',...
255     'vel_e','vel_f')
```

```
%**** to load existing presorted / calculated file begin here****
%comment out section of code starting with **C** above
%**D**if brdc files were sorted and velocity calculations not made ...
   above
%begin comment out this section
%{
if setnum == 1;
    load velocities1.mat
elseif setnum == 2;
    load velocities2.mat
elseif setnum == 3;
    load velocities3.mat
elseif setnum == 4;
    load velocities4.mat
elseif setnum == 5;
    load velocities5.mat
end

%extract data from orbit file
xv_a = vel_a(1,:);
yv_a = vel_a(2,:);
zv_a = vel_a(3,:);
timev_a = vel_a(4,:);
xv_b = vel_b(1,:);
yv_b = vel_b(2,:);
zv_b = vel_b(3,:);
timev_b = vel_b(4,:);
xv_c = vel_c(1,:);
yv_c = vel_c(2,:);
zv_c = vel_c(3,:);
timev_c = vel_c(4,:);
xv_d = vel_d(1,:);
yv_d = vel_d(2,:);
zv_d = vel_d(3,:);
timev_d = vel_d(4,:);
xv_e = vel_e(1,:);
yv_e = vel_e(2,:);
zv_e = vel_e(3,:);
timev_e = vel_e(4,:);
xv_f = vel_f(1,:);
yv_f = vel_f(2,:);
zv_f = vel_f(3,:);
timev_f = vel_f(4,:);
%}
%***comment out beginning at **D** if velocities are calculated in ...
   this run.
%this is the end of the section that loads existing data

%Build a matrix with position and velocity values matching times
%format of time, xp, yp, zp, xv, yv, zv
disp 'Building matrices of same time position and velocities ECEF'
[d_a] = compdyn(orbit_a,vel_a);
disp 'checkpoint A4'
```

```
    [d_b] = compdyn(orbit_b,vel_b);
310 disp 'checkpoint B4'
    [d_c] = compdyn(orbit_c,vel_c);
    disp 'checkpoint C4'
    [d_d] = compdyn(orbit_d,vel_d);
    disp 'checkpoint D4'
315 [d_e] = compdyn(orbit_e,vel_e);
    disp 'checkpoint E4'
    [d_f] = compdyn(orbit_f,vel_f);
    disp 'checkpoint F4'
    %these dynamics values are in the ECEF frame
320
    %compute greenwhich apparent siderial time angle
    [theta_g] = GAST(d_a(:,1));
    d_a (:,8) = theta_g;
    [theta_g] = GAST(d_b(:,1));
325 d_b (:,8) = theta_g;
    [theta_g] = GAST(d_c(:,1));
    d_c (:,8) = theta_g;
    [theta_g] = GAST(d_d(:,1));
    d_d (:,8) = theta_g;
330 [theta_g] = GAST(d_e(:,1));
    d_e (:,8) = theta_g;
    [theta_g] = GAST(d_f(:,1));
    d_f (:,8) = theta_g;
    disp 'Apparent siderial times calculated'
335
    %Calculate the dynamics variables in the ECI frame
    %xp, yp, zp, xv, yv, zv
    %A orbital plane
    d_a(:,9) = cosd(d_a(:,8)).*d_a(:,2);
340 d_a(:,10) = cosd(d_a(:,8)).*d_a(:,3);
    d_a(:,11) = d_a(:,4);
    d_a(:,12) = cosd(d_a(:,8)).*d_a(:,5);
    d_a(:,13) = cosd(d_a(:,8)).*d_a(:,6);
    d_a(:,14) = d_a(:,7);
345 %B orbital plane
    d_b(:,9) = cosd(d_b(:,8)).*d_b(:,2);
    d_b(:,10) = cosd(d_b(:,8)).*d_b(:,3);
    d_b(:,11) = d_b(:,4);
    d_b(:,12) = cosd(d_b(:,8)).*d_b(:,5);
350 d_b(:,13) = cosd(d_b(:,8)).*d_b(:,6);
    d_b(:,14) = d_b(:,7);
    %C orbital plane
    d_c(:,9) = cosd(d_c(:,8)).*d_c(:,2);
    d_c(:,10) = cosd(d_c(:,8)).*d_c(:,3);
355 d_c(:,11) = d_c(:,4);
    d_c(:,12) = cosd(d_c(:,8)).*d_c(:,5);
    d_c(:,13) = cosd(d_c(:,8)).*d_c(:,6);
    d_c(:,14) = d_c(:,7);
    %D orbital plane
360 d_d(:,9) = cosd(d_d(:,8)).*d_d(:,2);
    d_d(:,10) = cosd(d_d(:,8)).*d_d(:,3);
    d_d(:,11) = d_d(:,4);
    d_d(:,12) = cosd(d_d(:,8)).*d_d(:,5);
```

```
    d_d(:,13) = cosd(d_d(:,8)).*d_d(:,6);
365 d_d(:,14) = d_d(:,7);
    %E orbital plane
    d_e(:,9) = cosd(d_e(:,8)).*d_e(:,2);
    d_e(:,10) = cosd(d_e(:,8)).*d_e(:,3);
    d_e(:,11) = d_e(:,4);
370 d_e(:,12) = cosd(d_e(:,8)).*d_e(:,5);
    d_e(:,13) = cosd(d_e(:,8)).*d_e(:,6);
    d_e(:,14) = d_e(:,7);
    %F orbital plane
    d_f(:,9) = cosd(d_f(:,8)).*d_f(:,2);
375 d_f(:,10) = cosd(d_f(:,8)).*d_f(:,3);
    d_f(:,11) = d_f(:,4);
    d_f(:,12) = cosd(d_f(:,8)).*d_f(:,5);
    d_f(:,13) = cosd(d_f(:,8)).*d_f(:,6);
    d_f(:,14) = d_f(:,7);
380 disp 'Dynamics in ECI calculated'

    %calculation of moment values px py pz in ECI frame
    %A orbital plane
    d_a(:,15) = d_a(:,12) - omega.*d_a(:,10);
385 d_a(:,16) = d_a(:,13) - omega.*d_a(:,9);
    d_a(:,17) = d_a(:,14);
    %B orbital plane
    d_b(:,15) = d_b(:,12) - omega.*d_b(:,10);
    d_b(:,16) = d_b(:,13) - omega.*d_b(:,9);
390 d_b(:,17) = d_b(:,14);
    %C orbital plane
    d_c(:,15) = d_c(:,12) - omega.*d_c(:,10);
    d_c(:,16) = d_c(:,13) - omega.*d_c(:,9);
    d_c(:,17) = d_c(:,14);
395 %D orbital plane
    d_d(:,15) = d_d(:,12) - omega.*d_d(:,10);
    d_d(:,16) = d_d(:,13) - omega.*d_d(:,9);
    d_d(:,17) = d_d(:,14);
    %E orbital plane
400 d_e(:,15) = d_e(:,12) - omega.*d_e(:,10);
    d_e(:,16) = d_e(:,13) - omega.*d_e(:,9);
    d_e(:,17) = d_e(:,14);
    %F orbital plane
    d_f(:,15) = d_f(:,12) - omega.*d_f(:,10);
405 d_f(:,16) = d_f(:,13) - omega.*d_f(:,9);
    d_f(:,17) = d_f(:,14);

    %create canonical units matrix (will be used in freqident program)
    %positions x,y,z in ECEF (TU) and velocities x,y,z in ECI (DU/TU)
410 %A orbital plane
    c_a(:,1) = d_a(:,2).*(Re_c/Re);
    c_a(:,2) = d_a(:,3).*(Re_c/Re);
    c_a(:,3) = d_a(:,4).*(Re_c/Re);
    c_a(:,4) = d_a(:,12).*(su_c/su);
415 c_a(:,5) = d_a(:,13).*(su_c/su);
    c_a(:,6) = d_a(:,14).*(su_c/su);
    %B orbital plane
    c_b(:,1) = d_b(:,2).*(Re_c/Re);
```

```
c_b(:,2) = d_b(:,3).*(Re_c/Re);
c_b(:,3) = d_b(:,4).*(Re_c/Re);
c_b(:,4) = d_b(:,12).*(su_c/su);
c_b(:,5) = d_b(:,13).*(su_c/su);
c_b(:,6) = d_b(:,14).*(su_c/su);
%C orbital plane
c_c(:,1) = d_c(:,2).*(Re_c/Re);
c_c(:,2) = d_c(:,3).*(Re_c/Re);
c_c(:,3) = d_c(:,4).*(Re_c/Re);
c_c(:,4) = d_c(:,12).*(su_c/su);
c_c(:,5) = d_c(:,13).*(su_c/su);
c_c(:,6) = d_c(:,14).*(su_c/su);
%D orbital plane
c_d(:,1) = d_d(:,2).*(Re_c/Re);
c_d(:,2) = d_d(:,3).*(Re_c/Re);
c_d(:,3) = d_d(:,4).*(Re_c/Re);
c_d(:,4) = d_d(:,12).*(su_c/su);
c_d(:,5) = d_d(:,13).*(su_c/su);
c_d(:,6) = d_d(:,14).*(su_c/su);
%E orbital plane
c_e(:,1) = d_e(:,2).*(Re_c/Re);
c_e(:,2) = d_e(:,3).*(Re_c/Re);
c_e(:,3) = d_e(:,4).*(Re_c/Re);
c_e(:,4) = d_e(:,12).*(su_c/su);
c_e(:,5) = d_e(:,13).*(su_c/su);
c_e(:,6) = d_e(:,14).*(su_c/su);
%F orbital plane
c_f(:,1) = d_f(:,2).*(Re_c/Re);
c_f(:,2) = d_f(:,3).*(Re_c/Re);
c_f(:,3) = d_f(:,4).*(Re_c/Re);
c_f(:,4) = d_f(:,12).*(su_c/su);
c_f(:,5) = d_f(:,13).*(su_c/su);
c_f(:,6) = d_f(:,14).*(su_c/su);
disp 'canonical matrices complete'

save (filename5,'c_a','c_b','c_c','c_d',...
    'c_e','c_f')

%{
%******* THIS is code from earlier version ******

orbit = [xp;yp;zp];

%Plot x,y,z values
surfl(orbit);
shading interp
colormap(winter);
title('Position of Satellite in Earth Centered Earth Fixed frame')
xlabel('x position (km)')
ylabel('y position (km)')
zlabel('z position (km)')

lx=length(xp); %length of position vectors
%}
```

C.2 Function for Getting Positions from Precise Orbit Data (sp3 file)

Listing C.2: Satellite Positions from Precise Orbit Data

```
function [xp, yp, zp, time] = pos_final(PRN, ...
    filename)

%{
This function will pull the required data from GPS final \
satellite orbit data (as a combined sp3 file) The inputs are
the filename for the orbit data and the satellite to be analyzed
(PRN). It will out put the date/time information as a juliandate
and the positions x y z
%}

if PRN == 0;
    disp 'no satellite identified'
end

%build vehicle identification string
str = num2str(PRN,'%02d');
vehID = strcat('PG',str); %vehicle ID

%Read in the final orbit data(sp3 file)
fid = fopen(filename);
if fid == -1;
    disp 'error file can not be opened'
end
%determine the file length
first_ch = textscan(fid, '%s%*[^\n]');
fclose(fid);
file_len = length(first_ch{1});

%reopen the file and to pull out required data
fid = fopen(filename);

i = 1; %initialize time matrix
j = 1; %initialize position matrices
for n = 1:file_len
    tline = fgetl(fid);
    %scan file for date time stamp write to file
    if tline(1) == '*';
        %Yr, Mo, Day, hr, min, sec
        yr = str2double(tline(4:7));
        mo = str2double(tline(9:10));
        day = str2double(tline(12:13));
        hr = str2double(tline(15:16));
        min = str2double(tline(18:19));
        sec = str2double(tline(21:31));
        date = [yr, mo, day,hr,min,sec];
        time(i) = juliandate(date);%date time stamp
        %build dummy matrix if no satellite was identified
        if PRN == 0;
            xp(i) = 1;
            yp(i) = 1;
```

```
            zp(i) = 1;
        end
        i=i+1;
        %find PRN for the date and time write to file
    else if tline(1:4) == vehID;
        %x y z coordinates (km)
        xp(j) = str2double(tline(5:18));
        yp(j) = str2double(tline(19:32));
        zp(j) = str2double(tline(33:46));
        j=j+1;
        else
        end
    end
end
fclose(fid)

%check vector lengths
lx = length(xp); %x y z will all have the same length
lt = length(time);
if lx ~= lt;
    disp 'error vectors are not the same length'
end

end
```

C.3 Function for Calculating Velocities Based on Broadcast Ephemeris Data (07n file)

Listing C.3: Calculation of Velocities Based on Broadcast Ephemeris Data

```
function [xv, yv, zv, time] = vel_brdc(PRN, filename)

%{
This function will calculate the required data from GPS broadcast ...
   ephemeris
files. The inputs are the satellite to be analyzed(PRN) and the file...
    of the
broadcast ephemeris data. It will out put the date/time information ...
   as a
juliandate and the velocities x y z
%}

%This code is based on C code by Benjamin W Remondi
%reference ICD-200

format long e

%constants
mu = 3.986005e14; %m^3/s^2
omega_e = 7.2921151467e-5; %rad/s

%set string for satellite
if PRN <=9 ;
```

```
        str = [' ',num2str(PRN,'%2.0d')];
    else
        str = num2str(PRN,'%2d');
    end

    mil = num2str(20); %century

    %Read in the broadcast ephemeris data(07n file)
    fid = fopen(filename);
    if fid == -1;
        disp 'error file can not be opened'
    end
    %determine the file length
    first_ch = textscan(fid, '%s%*[^\n]');
    fclose(fid);
    file_len = length(first_ch{1})+1000;

    %reopen the file and to pull out required data
    fid = fopen(filename);

    i = 1; %initialize  matrices
    for n = 1:file_len %480
        tline = fgetl(fid); %line 1 of data set
        %scan file PRN value
        if length(tline) > 1 && strcmp(tline(1:2),str)==1;
            %Yr, Mo, Day, hr, min, sec
            yrstr = strcat(mil,tline(4:5));
            yr = str2double(yrstr);
            mo = str2double(tline(7:8));
            day = str2double(tline(10:11));
            hr = str2double(tline(13:14));
            min = str2double(tline(16:17));
            sec = str2double(tline(19:22));
            date = [yr, mo, day,hr,min,sec];
            time(i) = juliandate(date);%date time stamp
            %line 2 of data set
            tline = fgetl(fid);
            %amlitude of the sine harmonic correction term to orbit ...
               radius
            crs(i) = str2double(tline(23:41)); %meters
            %mean motion difference from computed value
            delta_n(i) = str2double(tline(42:60)); %rad/sec
            %mean anomaly at reference time
            m0(i) = str2double(tline(61:79)); %rad
            %line 3 of data set
            tline = fgetl(fid);
            %amlitude of the cosine harmonic correcyin term to Argument ...
               of
            %Latitude
            cuc(i) = str2double(tline(4:22)); %rad
            %eccentricity
            e(i) = str2double(tline(23:41));
            %amlitude of the sine harmonic correction term to Argument ...
               of
            %Latitude
```

```
        cus(i) = str2double(tline(42:60)); %rad
        %square root of semi-major axis
75      roota(i) = str2double(tline(61:79)); %sqrt(m)
        %line 4 of data set
        tline = fgetl(fid);
        %time of epoch
        toe(i) = str2double(tline(4:22)); %GPS wk sec
80      %amlitude of the cosine harmonic correction term to ...
           inclination
        cic(i) = str2double(tline(23:41)); %rad
        %lobgitude of the ascending node of orbital plane at weekly ...
           epoch
        bigomega0(i) = str2double(tline(42:60)); %rad
        %amlitude of the sine harmonic correction term to ...
           inclination
85      cis(i) = str2double(tline(61:79)); %rad
        %line 5 of data set
        tline = fgetl(fid);
        %inclination  angle at reference time
        i0(i) = str2double(tline(4:22)); %rad
90      %amlitude of the cosine harmonic correction term to orbit ...
           radius
        crc(i) = str2double(tline(23:41)); %meters
        %arguement of perigee
        smallomega(i) = str2double(tline(42:60)); %rad
        %Rate of right ascension
95      bigomegadot(i) = str2double(tline(61:79)); %rad/sec
        %line 6 of data set
        tline = fgetl(fid);
        %rate of inclination angle
        idot(i) = str2double(tline(4:22)); %rad
100     %convert day into day of the year
        if mo == 2;
            day = day + 31;
        elseif mo == 3;
            day = day + 59;
105     elseif mo == 4;
            day = day + 90;
        elseif mo == 5;
            day = day + 120;
        elseif mo == 6;
110         day = day + 151;
        elseif mo == 7;
            day = day + 181;
        elseif mo == 8;
            day = day + 212;
115     elseif mo == 9;
            day = day + 243;
        elseif mo == 10;
            day = day + 273;
        elseif mo == 11;
120         day = day + 304;
        elseif mo == 12;
            day = day + 334;
        end
```

```
            %calculation of GPS week second for given time
125         if day <= 6
                daysec = day*24*60*60;
                remainsec = hr*60*60 + min*60 + sec;
                wksec = daysec+remainsec;
            else
130             day = day + 1;
                while day > 7
                    day = day-7;
                end
                daysec = (day - 1)*24*60*60;
135             remainsec = hr*60*60 + min*60 + sec;
                wksec = daysec+remainsec;
            end
            t(i) = wksec; %GPS week seconds: time of pos & vel request
            i=i+1;
140     end
    end
    fclose(fid);

    %create dummy matrices if no satellite identified
145 if PRN == 0;
        disp 'no satellite identified'
        crs = ones(1,15000);
        delta_n = ones(1,15000);
        m0 = ones(1,15000);
150     cuc = ones(1,15000);
        e = ones(1,15000);
        cus = ones(1,15000);
        roota = ones(1,15000).*sqrt(26560000);
        toe = ones(1,15000);
155     cic = ones(1,15000);
        bigomega0 = ones(1,15000);
        cis = ones(1,15000);
        i0 = ones(1,15000)*55;
        crc = ones(1,15000);
160     smallomega = ones(1,15000);
        bigomegadot = ones(1,15000);
        idot = ones(1,15000);
        t = ones(1,15000)*1000;
        time = ones(1,15000);
165 end

    %begin calculations

    A = roota.^2; %semi-major axis
170 n0 = sqrt(mu./(A.^3)); %computed mean motion (rad/sec)
    n = n0+delta_n; %corrected mean motion
    tk = t - toe; %time from ephemeris reference epoch
    mk = m0+(n.*tk); %mean anomaly
    mkdot = n;
175 ek = mk;

    %keplers equation for eccentric anomaly
    for i = 1:10
```

```
    ek = mk+e.*sin(ek);
180 end

    ekdot = mkdot./(1.0-e.*cos(ek));

    nu = atan2((sqrt(1-e.^2).*sin(ek)),(cos(ek)-e)); %true anomaly
185 nudot = sin(ek).*ekdot.*(1.0+e.*cos(nu))./(sin(nu).*(1.0-e.*cos(ek))...
       );

    phik = nu+smallomega; %arguement of latitude
    %second harmonic perturbations
    corr_u = cus.*sin(2.*phik) + cuc.*cos(2.*phik); %Argument of ...
       Latitude correction
190 corr_r = crs.*sin(2.*phik) + crc.*cos(2.*phik); %Radius correction
    corr_i = cis.*sin(2.*phik) + cic.*cos(2.*phik); %Inclination ...
       correction

    uk = phik + corr_u; %corrected arguement of latitude
    rk = A.*(1-e.*cos(ek)) + corr_r; %corrected radius
195 ik = i0+idot.*tk + corr_i; %corrected inclination

    ukdot = nudot+2.*(cus.*cos(2*uk)-cuc.*sin(2*uk)).*nudot;
    rkdot = A.*e.*sin(ek).*n./(1-e.*cos(ek))+ ...
        2*(crs.*cos(2*uk)-crc.*sin(2*uk)).*nudot;
200 ikdot = idot +(cis.*cos(2*uk)-cic.*sin(2*uk)).*2.*nudot;

    %postions in orbital plane
    xpk = rk.*cos(uk);
    ypk = rk.*sin(uk);
205
    xpkdot = rkdot.*cos(uk)-ypk.*ukdot;
    ypkdot = rkdot.*sin(uk)+ xpk.*ukdot;

    %corrected logitude of the ascending node
210 omegak = bigomega0 + (bigomegadot-omega_e).*tk - omega_e.*toe;
    omegakdot = bigomegadot-omega_e;

    %earth-fixed coordinates
    xk = xpk.*cos(omegak) - ypk.*sin(omegak).*cos(ik);
215 yk = xpk.*sin(omegak) + ypk.*cos(omegak).*cos(ik);
    zk = ypk.*sin(ik);

    %velocities in m/s
    xkdot = (xpkdot-ypk.*cos(ik).*omegakdot).*cos(omegak) - ...
220     (xpk.*omegakdot+ypkdot.*cos(ik)-ypk.*sin(ik).*ikdot).*sin(omegak...
          );
    ykdot = (xpkdot-ypk.*cos(ik).*omegakdot).*sin(omegak) + ...
        (xpk.*omegakdot+ypkdot.*cos(ik)-ypk.*sin(ik).*ikdot).*cos(omegak...
          );
    zkdot =ypkdot.*sin(ik)+ypk.*cos(ik).*ikdot;

225 %velocities in km/s
    xv = xkdot.*0.001;
    yv = ykdot.*0.001;
    zv = zkdot.*0.001;
```

```
230
    %check vector lengths
    lxv = length(xv); %x y z will all have the same length
    lt = length(time);
    if lxv ~= lt;
235     disp 'error vectors are not the same length'
    end


    end
```

C.4 Function for Plotting the Frequencies and Identifying the Peaks

Listing C.4: Frequency and Power Plotting

```
    function [mFreqx, mFreqy, mFreqz] = pfplot(Plane, PRN, Yx, Yy, Yz)

    %{
    This function will plot the power and frequency for the orbital fft
  5 %}

    %general values and main figure
    nyquist = 1/2;
    titlestr = ['GPS',' ',Plane,' Orbital Plane: PRN =',' ',PRN];
 10 figure('Name',titlestr,'NumberTitle','off')

    %X position graph
    n = length(Yx);
    power_x = abs(Yx(1:(n/2))).^2;
 15 freq_x = (1:n/2)/(n/2)*nyquist;
    subplot(3,1,1)
    semilogy(freq_x,power_x)
    title ({titlestr;' ';'X position'})
    xlabel ('Frequency (orbits/15min)')
 20 ylabel('Power |Y(f)|')
    %find peak frequencies
    if strcmp(PRN, '00') == 1; % dummy if no satellite was identified
        hold on;
        index = find(power_x == max(power_x));
 25     mFreqx = num2str(freq_x(index));
        plot(freq_x(index),power_x(index),'r.','MarkerSize',10);
        tstr = ['\omega_1 =',' ',mFreqx];
        text(freq_x(index+50),power_x(index),tstr);
        hold off;
 30 elseif strcmp(PRN, '00') == 0;
        hold on;
        index = find(power_x == max(power_x(1:400)));
        mFreqx = num2str(freq_x(index));
        plot(freq_x(index),power_x(index),'r.','MarkerSize',10);
 35     tstr = ['\omega_1 =',' ',mFreqx];
        text(freq_x(index+50),power_x(index),tstr);
        index = find(power_x == max(power_x(400:900)));
```

```
        mFreqx2 = num2str(freq_x(index));
        plot(freq_x(index),power_x(index),'r.','MarkerSize',10);
        tstr = ['\omega_2 =',' ',mFreqx2];
        text(freq_x(index+50),power_x(index),tstr);
        index = find(power_x == max(power_x(900:end)));
        mFreqx3 = num2str(freq_x(index));
        plot(freq_x(index),power_x(index),'r.','MarkerSize',10);
        tstr = ['\omega_3 =',' ',mFreqx3];
        text(freq_x(index+50),power_x(index),tstr);
        hold off;
    end


    %Y position graph
    n = length(Yy);
    power_y = abs(Yy(1:(n/2))).^2;
    freq_y = (1:n/2)/(n/2)*nyquist;
    subplot(3,1,2)
    semilogy(freq_y,power_y)
    title ('Y position')
    xlabel ('Frequency (orbits/15min)')
    ylabel('Power |Y(f)|')
    %find peak frequencies
    if strcmp(PRN, '00') == 1; % dummy if no satellite was identified
        hold on;
        index = find(power_y == max(power_y));
        mFreqy = num2str(freq_y(index));
        plot(freq_y(index),power_y(index),'r.','MarkerSize',10);
        tstr = ['\omega_1 =',' ',mFreqy];
        text(freq_y(index+50),power_y(index),tstr);
        hold off;
    elseif strcmp(PRN, '00') == 0;
        hold on;
        index = find(power_y == max(power_y(1:400)));
        mFreqy = num2str(freq_y(index));
        plot(freq_y(index),power_y(index),'r.','MarkerSize',10);
        tstr = ['\omega_1 =',' ',mFreqy];
        text(freq_y(index+50),power_y(index),tstr);
        index = find(power_y == max(power_y(400:900)));
        mFreqy2 = num2str(freq_y(index));
        plot(freq_y(index),power_y(index),'r.','MarkerSize',10);
        tstr = ['\omega_2 =',' ',mFreqy2];
        text(freq_y(index+50),power_y(index),tstr);
        index = find(power_y == max(power_y(900:end)));
        mFreqy3 = num2str(freq_y(index));
        plot(freq_y(index),power_y(index),'r.','MarkerSize',10);
        tstr = ['\omega_3 =',' ',mFreqy3];
        text(freq_y(index+50),power_y(index),tstr);
        hold off;
    end

    %Z position graph
    n = length(Yz);
    power_z = abs(Yz(1:(n/2))).^2;
    freq_z = (1:n/2)/(n/2)*nyquist;
```

```
    subplot(3,1,3)
    semilogy(freq_z,power_z)
    title ('Z position')
    xlabel ('Frequency (orbits/15min)')
    ylabel('Power |Y(f)|')
    %find peak frequencies
    if strcmp(PRN, '00') == 1; % dummy if no satellite was identified
        hold on;
        index = find(power_z == max(power_z));
        mFreqz = num2str(freq_z(index));
        plot(freq_z(index),power_z(index),'r.','MarkerSize',10);
        tstr = ['\omega_1 =',' ',mFreqz];
        text(freq_z(index+50),power_z(index),tstr);
        hold off;
    elseif strcmp(PRN, '00') == 0;
        hold on;
        index = find(power_z == max(power_z(1:500)));
        mFreqz = num2str(freq_z(index));
        plot(freq_z(index),power_z(index),'r.','MarkerSize',10);
        tstr = ['\omega_1 =',' ',mFreqz];
        text(freq_z(index+50),power_z(index),tstr);
        index = find(power_z == max(power_z(500:1090)));
        mFreqz2 = num2str(freq_z(index));
        plot(freq_z(index),power_z(index),'r.','MarkerSize',10);
        tstr = ['\omega_2 =',' ',mFreqz2];
        text(freq_z(index+50),power_z(index),tstr);
        index = find(power_z == max(power_z(1090:end)));
        mFreqz3 = num2str(freq_z(index));
        plot(freq_z(index),power_z(index),'r.','MarkerSize',10);
        tstr = ['\omega_3 =',' ',mFreqz3];
        text(freq_z(index+50),power_z(index),tstr);
        hold off;
    end

    %display all frequencies
    format long e
    if strcmp(PRN, '00') == 0;
        x1 = num2str(mFreqx);
        x2 = num2str(mFreqx2);
        x3 = num2str(mFreqx3);
        y1 = num2str(mFreqy);
        y2 = num2str(mFreqy2);
        y3 = num2str(mFreqy3);
        z1 = num2str(mFreqz);
        z2 = num2str(mFreqz2);
        z3 = num2str(mFreqz3);
        display ([Plane,' Orbital Plane Frequencies are as follows:']);
        display (['x1 =', x1]);
        display (['x2 =', x2]);
        display (['x3 =', x3]);
        display (['y1 =', y1]);
        display (['y2 =', y2]);
        display (['y3 =', y3]);
        display (['z1 =', z1]);
        display (['z2 =', z2]);
```

```
        display (['z3 =', z3]);
    end
150
    end
```

C.5 Function for Computing the Greenwich Apparent Sidereal Time Angle

Listing C.5: Greenwich Apparent Sidereal Time Angle

```
function [theta_g] = GAST(jd)

%this function will caluculate...

% Using the USNO guidlines found at
% http://aa.usno.navy.mil/faq/docs/GAST.php,
% and using the "alternative formula" that can be used with a loss ...
   of
% precision of 0.1 second per century.

%The Naval Observatory can display Apparent Sideral Time given an ...
   input
%longitude directly as a comparison:
%http://tycho.usno.navy.mil/sidereal.html

D = jd - 2451545.0;
GMST = (18.697374558 + 24.06570982441908.*D) - ...
    24*floor((18.697374558 + 24.06570982441908.*D)./24);
        %greenwhich mean sidereal time, wrapped to [0 24) hours
omega = 125.04-0.052954.*D;
L = 280.47+0.98565.*D;
epsilon = 23.4393-0.0000004.*D;
deltapsi = -0.000319.*sind(omega)-0.000024.*sind(2*L);
eqeq = deltapsi.*cosd(epsilon);
GAST = GMST+eqeq;

theta_g = zero22pi(GAST*360/24);  %greenwhich meridian angle, in ...
   degrees

end
```

C.6 Function for Computing Dynamics

Listing C.6: Satellite Dynamics Calculations

```
function [dynamics] = compdyn(orbit,vel)

%This function will compare & combine the dynamics information ...
   avialable

%Build a matrix with position and velocity values matching times
%also correct julian date (time) to be based on UTC rather than GPS
%GPS was set to UTC 6 Jan 1980 and does not include leap seconds
```

```
%positon components
xp = orbit(1,:);
yp = orbit(2,:);
zp = orbit(3,:);
timep = orbit(4,:);
%velocity components
xv = vel(1,:);
yv = vel(2,:);
zv = vel(3,:);
timev = vel(4,:);

lp = length(timep);
lv = length(timev);

%initialization
i = 1;
j = 1;
k = 1;

while i < lp+1;
    while j < lv+1;
        if timep(i) == timev(j);
            dynamics(k,1) = timep(i) + 0.00016;
            dynamics(k,2) = xp(i);
            dynamics(k,3) = yp(i);
            dynamics(k,4) = zp(i);
            dynamics(k,5) = xv(j);
            dynamics(k,6) = yv(j);
            dynamics(k,7) = zv(j);
            k = k+1;
            i = i+1;
            j = j+1;
        elseif timep(i) > timev(j);
            j = j+1;
        elseif timep(i) < timev(j);
            i = i+1;
        elseif i >= lp;
            j = lv+2;
            i = i+2;
        elseif j >= lv;
            i = lp+2;
            j = j+2;
        else
            i = i+1;
            j = j+1;
        end
    end
    if j >= lv;
        i = lp + 2;
        j = j + 2;
    end
end

%if no satellite was identified, create a dummy output matrix
```

```
   if xp == zp;
      display 'no satellite identified'
65    dynamics = ones (15000,7);
   end

   end
```

C.7 Code to Merge Files for Analysis

Code is available on the web to merge precise data files. This code is unable to be automated and only combines two files at a time. Another problem with the current code is that it only provides the times of the data, but the merged file does not contain the satellite data. Because of these limitations a new code was created to combine the necessary parts of many data files.

Bibliography

Air Force Space Command (AFSPC). "Global Positioning System Factsheet," March 2007. March 2008. http://www.af.mil/factsheets/factsheet.asp?id=119.

ARINC Research Corporation. *Navstar GPS Space Segment / Navigation User Interfaces.*: GPS Navstar JPO, April 2000.

Arnold, V. "Proof of Kolmogorov's Theorem on the Preservation of Quasi-Periodic Motions Under Small Pertubations of the Hamiltonian," *Rus. Math Surv. ,18*(N6) (1963).

Bate, R. R., D. D. Mueller, and J. E. White. *Fundamentals of Astrodynamics.* New York: Dover Publications, 1971.

Bordner, R. "GPS SVN 25 Status," 7 March 2008. Personal Correspondence.

Campbell, W. H. *Introduction to Geomagnetic Fields.* New York: Cambridge Univesity Press,, 2003.

Celletti, A. "KAM Stability and Celestial Mechanics," *Planet Space Sci ,46*(11/12), 1433 (1998).

Celletti, A. "KAM Tori for N-body Problems: a Brief History," *Celestial Mech.Dynam.Astronom. ,95*(1-4), 117 (2006).

Celletti, A., and L. Chierchia. "KAM Stability for a Three-Body Problem of the Solar System," *Z. Angew. Math. Phys. ,57*(1), 33–41 (2005).

Celletti, A., C. Froeschl, and E. Lega. "Frequency Analysis of the Stability of Asteroids in the Framework of the Restricted Three-Body Problem," *Celestial Mech.Dynam.Astronom. ,90*(3-4), 245 (2004).

Gurtner, W. "RINEX: The Receiver Independent Exchange Format Version 2.10," 25 January 2002.

Hilla, S. "The Extended Standard Product 3 Orbit Format (SP3-c)," *National Geodetic Survey* (2007).

International GNSS Service (IGS). "IGS Products," 21 Nov 2005. 15 Oct 2007. http://igscb.jpl.nasa.gov/components/prods.html.

International GNSS Service (IGS). "Data and Products; CDDIS," 1 Mar 2007. 9 Jan 2008. http://igscb.jpl.nasa.gov/components/dcnav/cddis_products_wwww.html , ftp://cddis.gsfc.nasa.gov/.

Kolmogorov, A. "On the Conservation of Conditionally Periodic Motions Under Small Pertubations of the Hamiltonian," *Dokl. Akad. Nauk.98*, 469 (January 1954).

Laskar, J. "Introduction to Frequency Map Analysis," *Hamiltonian Systems with Three or More Degrees of Freedom*p. 134 (1999).

Laskar, J. "Frequency Map Analysis and Quasiperiodic Decompositions," 2003. Porquerolles School.

McGill, C., and J. Binney. "Torus Construction in General Gravitational Potentials," *Monthy Notices Royal Astronomical Society ,244*, 634 (Jan 1990).

Milcom Monitoring Post. "USNO Block II GPS Satellite Information," Jun 2007. 5 Jan 2008. http://mt-milcom.blogspot.com/2007/06/usno-block-ii-gps-satellite-information.html.

Misra, P., and P. Enge. *Global Positioning System : Signals, Measurements, and Performance*. Lincoln, Mass.: Ganga-Jamuna Press, 2001.

Moser, J. "On Invariant Curves of Area-Preserving Mappings of an Annulus," *Nachr.Akad.Wiss.Gttingen Math.-Phys.Kl.II ,Kl*(IIa), 1–20 (1962).

National Aeronautics and Space Administration (NASA). "EGM 96 The NASA GSFC and NIMA Joint Geopotential Model," 1998. 15 January 2008. http://cddis.nasa.gov/926/egm96/egm96.html.

National Aeronautics and Space Administration (NASA). "Orbital Debris Quarterly News," *12*(1) (January 2008).

National Geodetic Survey (NGS). "2007 GPS Calendar," 2007a. 15 Oct 2007. http://www.ngs.noaa.gov/CORS/gpscal07.html.

National Geodetic Survey (NGS). "National Geodetic Survey, Precise GPS Orbits," 27 Aug 2007b. 15 Oct 2007. http://www.ngs.noaa.gov/orbits/.

National Geospatial Intelligence Agency (NGIA). "Current GPS Satellite Data," 07 January 2008. 11 Jan 2008. http://earth-info.nima.mil/GandG/sathtml/satinfo.html.

Remondi, B. "Computing Satellite Velocity Using the Broadcast Ephemeris," *GPS Solutions, 8*(3), 181–183 (2004).

Remondi, B. "Computing Satellite Velocity using the Broadcast Ephemeris, C code," 2004b. 11 Jan 2008. http://www.ngs.noaa.gov/gps−toolbox/bc_velo.html.

United States Department of Commerce, National Oceanic and Space Administration (NOAA). "Halloween Space Weather Storms of 2003," *NOAA Technical Memorandum OAR SEC-88* (June 2004).

US Congress Office of Technology Assessment (OTA). "Orbiting Debris: A Space Environmental Problem - Background Paper," *OTA-BP-ISC-72* (September 1990).

US Naval Observatory (USNO). "Multiyear Interactive Computer Almanac," 2006. 5 Febuary 2008. http://aa.usno.navy.mil/software/mica/micainfo.php.

US Naval Observatory (USNO). "Approximate Sidereal Time," . 5 Feb 2008. http://aa.usno.navy.mil/faq/docs/GAST.php.

Wiesel, W. E. *Modern Astrodynamics*. Beavercreek, OH: Aphelion Press, 2003.

Wiesel, W. E. "Earth Satellite Orbits as KAM Tori," AAS/AIAA Astrodynamics Specialist Conference. 19-23 Aug 2007. AAS 07-423.

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
27–03–2008	Master's Thesis	Mar 2007 — Mar 2008

4. TITLE AND SUBTITLE

Modeling GPS Satellite Orbits Using KAM Tori

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Rachel M. Derbis, Capt, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GA/ENY/08-M09

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

N/A

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Global Positioning System (GPS) satellite orbits are modeled using Kolmogorov, Arnold, Moser (KAM) tori. Precise Global Positioning System satellite locations are analyzed using Fourier transforms to identify the three basis frequencies in an Earth Centered, Earth Fixed (ECEF) rotating reference frame. The three fundamental frequencies are 1) the anomalistic frequency, 2) a combination of earth's rotational frequency and the nodal regression rate, and 3) the apsidial regression rate. A KAM tori model fit to the satellite data could be used to predict future satellite locations. This model would allow rapid determination with fewer computational requirements than the typical method of integrating through an orbit.

15. SUBJECT TERMS

Orbiting Satellites, Global Positioning System, Torus Action on Euclidean Space, Fast Fourier Transforms, Kolmogorov Arnold Moser Tori, Kolmogorov Arnold Moser Torus, Orbit Modeling, Astrodynamics, Satellite Trajectory

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. William W Wiesel
U	U	U	UU	87	19b. TELEPHONE NUMBER *(Include area code)* (937) 255-3636 x 4312; william.wiesel@afit.edu